

**НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ БІОРЕСУРСІВ
І ПРИРОДОКОРИСТУВАННЯ УКРАЇНИ**

Факультет інформаційних технологій

ПОГОДЖЕНО
Декан факультету

_____ **Ігор БОЛБОТ**
(підпис)

«__» _____ 2026 р.

ДОПУСКАЄТЬСЯ ДО ЗАХИСТУ
Завідувач кафедри комп'ютерних наук

_____ **Белла ГОЛУБ**
(підпис)

«__» _____ 2026 р.

БАКАЛАВРСЬКА КВАЛІФІКАЦІЙНА РОБОТА
на тему: « Програмне забезпечення системи обліку та ідентифікації великої
рогатої худоби за дерматогліфами »

Спеціальність «121 Інженерія програмного забезпечення»

Освітньо-професійна програма «Інженерія програмного забезпечення»

Гарант освітньої програми

к.т.н., доцент

_____ **Ганна ВАЙГАНГ**
(підпис)

Керівник бакалаврської
кваліфікаційної роботи

професор, доктор економічних наук

_____ **Роман РУДЕНСЬКИЙ**
(підпис)

Виконав

_____ **Михайло МІХЛЕВСЬКИЙ**
(підпис)

Київ-2026

**НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ БІОРЕСУРСІВ
І ПРИРОДОКОРИСТУВАННЯ УКРАЇНИ**

Факультет інформаційних технологій

ЗАТВЕРДЖУЮ
Завідувач кафедри комп'ютерних наук

К.Т.Н., доцент _____ **Белла ГОЛУБ**
(підпис)
(підпис)

« ____ » _____ 2026 р.

ЗАВДАННЯ
ДО ВИКОНАННЯ БАКАЛАВРСЬКОЇ КВАЛІФІКАЦІЙНОЇ РОБОТИ
ЗДОБУВАЧУ

Міхлевському Михайлу Сергійовичу

Спеціальність «121 Інженерія програмного забезпечення»

Освітньо-професійна програма «Інженерія програмного забезпечення»

Тема бакалаврської кваліфікаційної роботи

«Програмне забезпечення системи обліку та ідентифікації великої рогатої худоби за дерматогліфами»

затверджена наказом від «02» квітня 2026 р. № 939 «С»

Термін подання завершеної роботи на кафедру «22» травня 2026 р.

Вихідні дані до бакалаврської кваліфікаційної роботи (проєкту): надання інформації про облік великої рогатої худоби на фермах у вигляді списків та дерматогліфів.

Перелік питань, що підлягають дослідженню:

- Аналіз проблемної області
- Моделювання предметної області
- Проєктування програмної системи
- Впровадження та експлуатація системи

Дата видачі завдання « ____ » _____ 2026 р.

**Керівник бакалаврської
кваліфікаційної роботи**

(підпис)

Роман РУДЕНСЬКИЙ

(підпис)

(підпис)

**Завдання взяв до
виконання**

(підпис)

Михайло МІХЛЕВСЬКИЙ

(підпис)

ЗМІСТ

ВСТУП	4
1 АНАЛІЗ ПРОБЛЕМНОЇ ОБЛАСТІ	6
1.1 Опис предметної області	6
1.2 Огляд існуючих рішень	9
1.3 Постановка завдання	14
1.4 Функціональні та нефункціональні вимоги	15
1.5 Вимоги до інтерфейсу користувача	17
2 МОДЕЛЮВАННЯ ПРЕДМЕТНОЇ ОБЛАСТІ	19
2.1 Загальні відомості	19
2.2 Об'єктне та функціональне моделювання	20
2.3 Абстракції предметної області	30
2.4 Діаграма класів	32
3 ПРОЄКТУВАННЯ ПРОГРАМНОЇ СИСТЕМИ	35
3.1 Логічна модель даних	35
3.2 Вибір системи управління базою даних та її реалізація	40
3.3 Архітектура програмного забезпечення	43
3.4 Організаційна структура програмного забезпечення	46
3.5 Вибір інструментарію для створення програмного забезпечення	48
4 ВПРОВАДЖЕННЯ ТА ЕКСПЛУАТАЦІЯ СИСТЕМИ	50
4.1 Вимоги до апаратного та програмного забезпечення	50
4.2 Тестування системи	52
ВИСНОВКИ	57
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ	59
ДОДАТОК А	62
ДОДАТОК Б	65
ДОДАТОК В	7

ВСТУП

Ідентифікація та реєстрація худоби відіграють вирішальну роль у сучасному сільському господарстві, забезпечуючи ефективне управління, моніторинг здоров'я, контроль розведення та відстеження тварин протягом усього їхнього життєвого циклу. Традиційні методи ідентифікації, такі як вушні бирки, таврування та RFID-чіпи, широко використовуються, але мають низку обмежень. Бирки можуть бути втрачені, пошкоджені або навмисно видалені; таврування викликає стрес та викликає занепокоєння щодо добробуту тварин; а електронні ідентифікатори є відносно дорогими та вимагають спеціалізованих зчитувачів.

Альтернативним, перспективним підходом до ідентифікації тварин є використання біометричних даних, зокрема дерматогліфів — унікальних візерунків шкіри, що знаходяться на носах великої рогатої худоби. Подібно до відбитків пальців людини, ці візерунки є індивідуальними та залишаються незмінними протягом життя тварини. Цей метод є неінвазивним, його важко підробити та можна реалізувати за допомогою технологій цифрової візуалізації.

У цій бакалаврській роботі представлено розробку веб-програмної системи, яка підтримує облік та ідентифікацію великої рогатої худоби за дерматогліфами. Система розроблена для використання фермерами, ветеринарами та регуляторними органами для реєстрації худоби, управління їхніми персональними даними та завантаження біометричних зображень з метою ідентифікації. Програмне рішення було реалізовано за допомогою фреймворку PHP Symfony з базою даних MySQL, що забезпечує гнучкість, продуктивність та безпеку.

Ефективна ідентифікація худоби є важливою для ведення точного обліку, забезпечення здоров'я тварин та запобігання крадіжкам або шахрайству в сільськогосподарському секторі. Традиційні методи

ідентифікації, такі як вушні бирки, брендування та RFID-чіпи, широко використовуються, але мають суттєві недоліки, включаючи ризик фальсифікації, випадкового видалення та обмежену точність з часом. Ці обмеження підкреслюють **актуальність** інноваційних, безпечних та стійких до несанкціонованого доступу методів ідентифікації.

Дерматогліфи — унікальні візерунки на шкірі, що знаходяться на носах великої рогатої худоби, — представляють собою біометричну альтернативу, яка є постійною, неінвазивною та біологічно індивідуальною. Оскільки цифрові технології стають більш доступними, розробка програмних систем, здатних зберігати та керувати такими біометричними даними, стає все більш актуальною. Інтеграція ідентифікації на

Метою цієї роботи є покращення сфери реєстрації худоби, управління її особистими даними та завантаження біометричних зображень за допомогою створення функціонального рішення, яке демонструє, як ідентифікацію на основі дерматогліфів можна інтегрувати в доступну систему управління худобою.

на основі дерматогліфів у системі обліку худоби не тільки покращує відстежуваність, але й підтримує прийняття рішень на основі даних в управлінні стадом.

Предметом дослідження є розробка програмних засобів та веб-технологій для впровадження системи ідентифікації тварин на основі біометричних дерматогліфічних даних.

Об'єктом дослідження є процес обліку та ідентифікації великої рогатої худоби в контексті управління сільським господарством та ветеринарією з використанням цифрових інформаційних систем.

Цей проєкт не лише пропонує практичне програмне рішення, але й досліджує потенціал біометричної ідентифікації в сільськогосподарській галузі. Майбутні вдосконалення можуть включати інтеграцію алгоритмів машинного навчання для автоматизації розпізнавання дерматогліфів та розширення можливостей ідентифікації системи.

1 АНАЛІЗ ПРОБЛЕМНОЇ ОБЛАСТІ

1.1 Опис предметної області

У сфері сучасного тваринництва точна ідентифікація та належний облік худоби мають вирішальне значення для підтримки простежуваності, управління програмами розведення, запобігання шахрайству та ефективного реагування на спалахи захворювань. Незважаючи на широке використання методів ідентифікації, таких як вушні бирки, таврування та електронні чіпи, ці традиційні методи часто не відповідають вимогам надійності, економічної ефективності та добробуту тварин.

Вушні бирки, хоча й широко використовуються завдяки низькій вартості та простоті застосування, схильні до втрати, пошкодження або навмисного видалення. Таврування, хоча й забезпечує його тривалість, є інвазивним та викликає етичні проблеми через стрес і дискомфорт, які воно завдає тваринам. RFID-чіпи та інші електронні системи забезпечують безпечнішу ідентифікацію, але вимагають дорогої інфраструктури та спеціалізованого обладнання, що робить їх недоступними для багатьох малих та середніх ферм [1].

Обмеження цих традиційних методів викликали інтерес до альтернативних підходів, зокрема тих, що стосуються біометрії. В останні роки дослідники та практики почали досліджувати використання унікальних біологічних особливостей тварин для ідентифікації. Одним з перспективних рішень є застосування дерматогліфіки — візерунків шкірних гребенів на носі корови, які, як і відбитки пальців людини, є унікальними та залишаються незмінними протягом усього життя тварини.

Ідентифікація на основі дерматогліфів — це неінвазивний та захищений від несанкціонованого доступу метод, який не потребує додаткових фізичних міток чи електронних пристроїв. Він вимагає лише отримання цифрового зображення відбитка носа корови, яке потім можна зберегти та

використовувати для майбутньої ідентифікації. Цей метод не тільки біологічно безпечний, але й можливий за допомогою стандартних інструментів візуалізації, що робить його привабливим варіантом для фермерів та керівників тваринництва, які шукають сучасне та етичне рішення [2].

Незважаючи на свій потенціал, практичне впровадження ідентифікації дерматогліфів у повсякденну сільськогосподарську діяльність залишається обмеженим. Помітно бракує зручних веб-платформ, які інтегрують біометричну ідентифікацію великої рогатої худоби з функціями обліку стада. Більшість існуючого програмного забезпечення для управління фермами зосереджені переважно на логістиці, такій як графіки годування, моніторинг ваги та ветеринарні записи, не пропонуючи підтримки для розширеної біометричної ідентифікації.

Ця прогалина підкреслює необхідність спеціалізованої програмної системи, яка може полегшити управління обліком великої рогатої худоби, одночасно включаючи ідентифікацію на основі відбитків носа. Рішення, побудоване на відкритих, широко впроваджених технологіях, таких як PHP (Symfony framework) та MySQL, може ефективно служити цій меті. Воно надасть користувачам централізовану платформу для реєстрації тварин, завантаження та керування біометричними зображеннями, а також ведення комплексних записів безпечним та доступним способом.

Актуальність цієї теми полягає в її потенціалі для впровадження інноваційних біометричних технологій у сектор, який дедалі більше покладається на інструменти, що базуються на даних, для сталого та ефективного управління худобою. Інтеграція ідентифікації дерматогліфів у веб-додаток не лише підвищує надійність, але й відкриває шлях до майбутніх удосконалень, таких як застосування штучного інтелекту для автоматичного розпізнавання образів.

Управління худобою є життєво важливим компонентом сільськогосподарських систем у всьому світі, а здатність точно враховувати та ідентифікувати окремих тварин відіграє центральну роль у забезпеченні

здоров'я, простежуваності та продуктивності стад. Традиційно фермери та ветеринарні спеціалісти покладалися на зовнішні ідентифікатори, такі як вушні бирки, клеймо та мікрочіпи, для розрізнення тварин. Однак ці методи часто ненадійні, схильні до втрати або підробки, а в деяких випадках викликають занепокоєння щодо добробуту тварин.

У відповідь на ці обмеження сільськогосподарський сектор почав досліджувати біометричні технології, які пропонують більш безпечні та гуманні варіанти ідентифікації. Одним особливо перспективним напрямком є використання дерматогліфів — складних візерунків на шкірі носа корови, також відомих як відбитки носа. Ці візерунки унікальні для кожної особини, не змінюються з часом і можуть бути зафіксовані за допомогою стандартного обладнання для візуалізації, не спричиняючи дискомфорту чи стресу тварині.

Предметом цього дослідження є розробка та впровадження інформаційної системи, яка використовує біометричну ідентифікацію на основі дерматогліфів та поєднує її з інструментами для управління даними про велику рогату худобу. Ця система задумана як веб-застосунок, що дозволяє користувачам реєструвати тварин, завантажувати та зберігати зображення їхніх відбитків носа, а також вести вичерпні записи, такі як порода, вік, стан здоров'я та дані про власника.

З технологічної точки зору, проєкт базується на інструментах з відкритим кодом, таких як PHP, з використанням фреймворку Symfony, у поєднанні з базою даних MySQL для надійного зберігання даних. Такий вибір технології забезпечує доступність, масштабованість та легкість інтеграції з майбутніми модулями, такими як алгоритми штучного інтелекту для автоматичного розпізнавання образів або аналіз даних для оптимізації стада.

Предметна галузь також перетинається з ширшими тенденціями в цифровому сільському господарстві та точному землеробстві, де технологічні інновації використовуються для підвищення ефективності та сталості сільськогосподарських процесів. Замінюючи застарілі або схильні до помилок методи ідентифікації біометричним рішенням, система сприяє точнішому

відстеженню, кращому контролю захворювань та ефективнішому управлінню ресурсами в тваринницьких операціях.

Підсумовуючи, ця предметна галузь об'єднує елементи біометрії тварин, систем баз даних, веб-технологій та сільськогосподарської інформатики для вирішення реальної проблеми в управлінні тваринництвом. Вона відображає перехід до інтелектуальних, етичних та заснованих на даних рішень у сучасній сільськогосподарській практиці [3].

1.2 Огляд існуючих рішень

У сфері ідентифікації та управління худобою протягом багатьох років було розроблено різні технологічні рішення для оптимізації процесів обліку та забезпечення надійної ідентифікації тварин. Ці системи варіюються від базових методів маркування до складних цифрових платформ. Однак більшість існуючих рішень спираються на зовнішні ідентифікатори або електронні пристрої та не включають біометричну ідентифікацію на основі унікальних анатомічних ознак, таких як дерматогліфи.

AgriWebb — це хмарна платформа для управління фермерським господарством, розроблена спеціально для виробників худоби. Засноване в Австралії, програмне забезпечення отримало міжнародне визнання завдяки своїй здатності оцифровувати та спрощувати складні процеси ведення обліку худоби, дотримання вимог та прийняття рішень. Його основна цінність полягає в перетворенні традиційних операцій з ручкою та папером на динамічний цифровий робочий процес у режимі реального часу, що дозволяє фермерам працювати ефективніше та приймати рішення на основі даних [4].

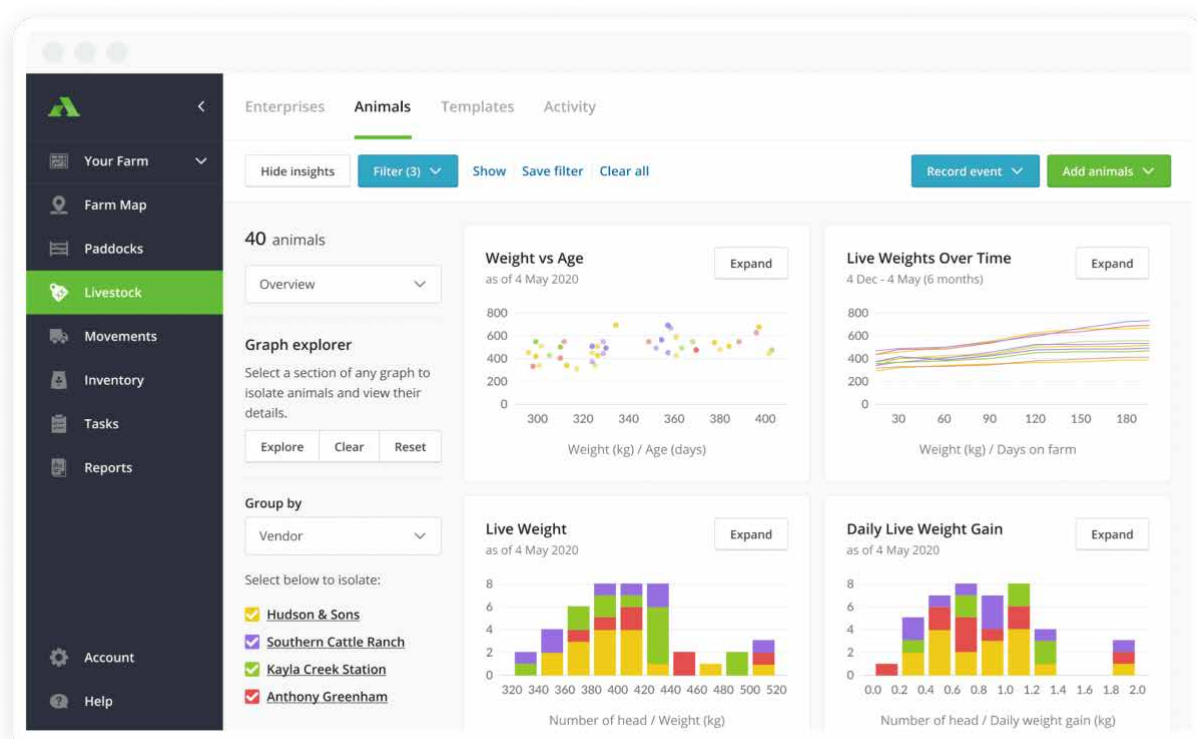


Рис. 1.1 Інформаційна система “AgriWebb”

Однією з ключових особливостей AgriWebb є підтримка як індивідуального управління тваринами, так і відстеження груп тварин. Ця подвійна функціональність робить її придатною для широкого спектру тваринницьких операцій — від малих ферм до великих комерційних підприємств. Користувачі можуть записувати дати народження, методи лікування, вагу, інформацію про розведення тощо для окремих тварин або цілих груп. Ця гнучкість особливо корисна для фермерів, які прагнуть масштабувати свою діяльність, не втрачаючи контролю над точністю записів.

Інтерактивна карта ферми AgriWebb — ще одна визначна функція. Вона дозволяє користувачам візуально організовувати свої пасовища, планувати ротацію випасу, контролювати використання пасовищ та призначати завдання. Карта підтримує оновлення в режимі реального часу та безперешкодно інтегрується з іншими модулями системи, допомагаючи фермерам оптимізувати використання землі та зменшити надмірний випас худоби. Візуальний характер цього інструменту полегшує всім членам

команди, незалежно від їхнього технічного досвіду, співпрацю та розуміння структури ферми [4].

Платформа також чудово справляється з управлінням завданнями та автоматизацією робочих процесів. Користувачі можуть створювати, призначати та контролювати завдання для всієї команди через централізовану панель інструментів. Незалежно від того, чи це перевірка водопоїв, вакцинація чи переміщення стад між загонами, AgriWebb гарантує, що критично важливі дії плануються, відстежуються та виконуються. Ця функція не тільки покращує підзвітність, але й допомагає зменшити ймовірність помилок та пропущених термінів.

Дотримання нормативних вимог та відстеження є ключовими компонентами сучасного управління тваринництвом, і AgriWebb надає комплексні інструменти для обох. Система дозволяє фермерам легко фіксувати обробку, переміщення та продажі, створюючи звіти, що відповідають галузевим стандартам та урядовим вимогам. В Австралії та інших країнах AgriWebb інтегрується з національними базами даних худоби, що спрощує подання необхідних даних та підтримку відповідності.

Мобільний додаток розроблено для використання в автономному режимі, що особливо важливо в сільській місцевості або віддалених районах з обмеженим доступом до Інтернету. Фермери можуть вводити дані в полі, навіть без підключення до Інтернету, та синхронізувати їх із хмарою після повернення до мережі. Ця функціональність гарантує, що жодна важлива інформація не буде втрачена, а вся діяльність ферми буде точно задокументована в режимі реального часу.

На завершення, AgriWebb – це потужний інструмент, який привносить інновації в традиційне тваринництво за допомогою цифрових технологій. Пропонуючи такі функції, як біометричне ведення обліку, управління завданнями, відстеження відповідності та аналітика – все в одній платформі, – він дозволяє фермерам ефективніше керувати своїми операціями. Хоча AgriWebb наразі не підтримує біометричну ідентифікацію за допомогою

дерматогліфів, його модульна структура передбачає потенціал для майбутньої інтеграції таких технологій, що робить його надійним орієнтиром для систем, спрямованих на перехід управління тваринництвом на наступне покоління.

Розглянемо інше програмне рішення на ринку.

FarmWizard – це комплексне програмне забезпечення для управління фермерським господарством, розроблене Orchid FarmTech, призначене для оптимізації операцій з тваринництвом та забезпечення дотримання сільськогосподарських норм. Засноване в середині 2000-х років, програмне забезпечення було розроблено у співпраці з FarmWizard з Белфаста для задоволення зростаючої потреби в хмарних рішеннях у сільському господарстві. Воно пропонує спеціалізовані модулі для молочного, м'ясного та вівчарського господарства, надаючи фермерам інструменти для управління обліком тварин, моніторингу продуктивності та забезпечення відповідності законодавчим вимогам [5].

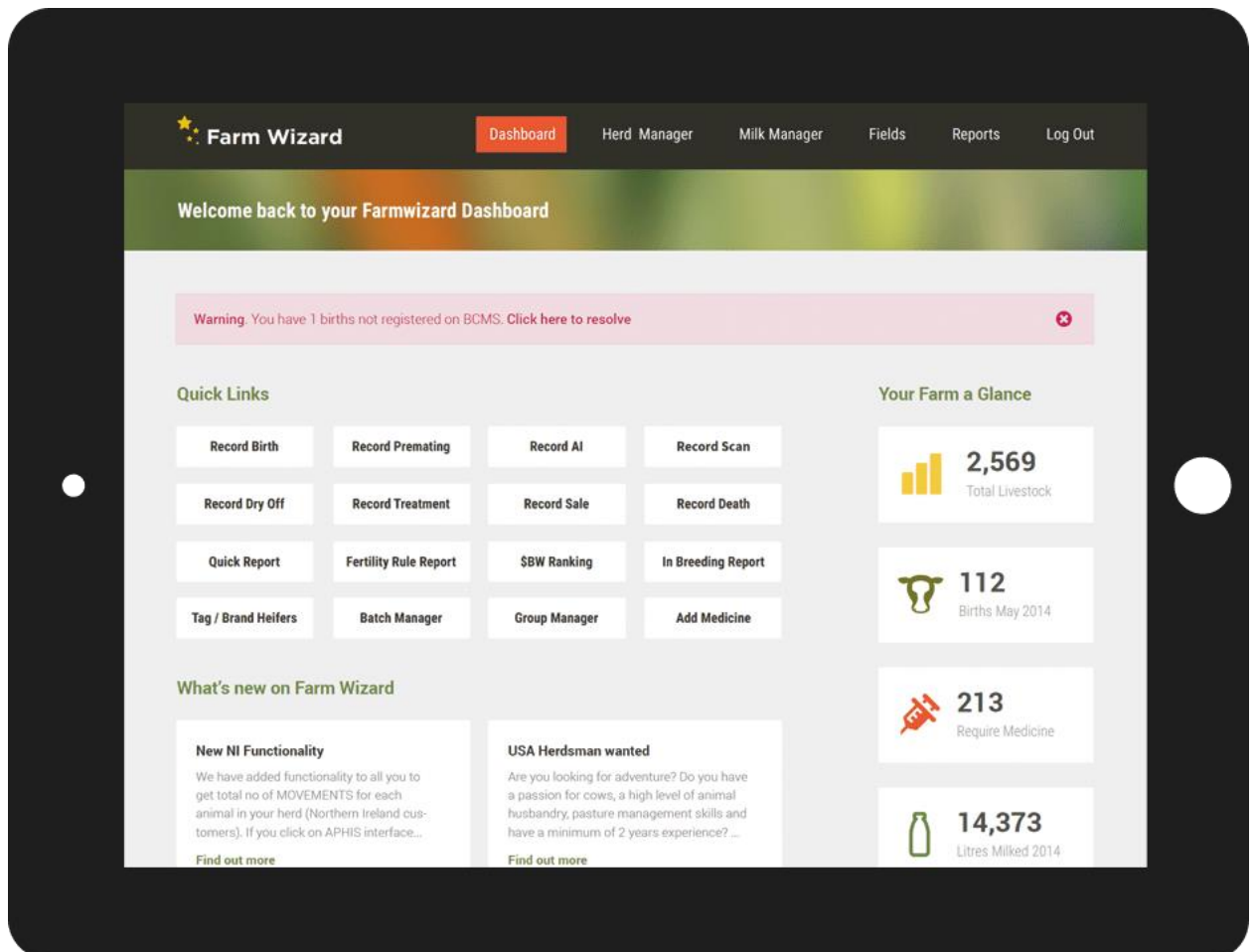


Рис. 1.2 Інформаційна система “FarmWizard”

Однією з видатних особливостей FarmWizard є його веб-архітектура, яка дозволяє фермерам отримувати доступ до системи з різних пристроїв, включаючи ПК, Mac, планшети та смартфони. Ця гнучкість гарантує, що користувачі можуть вводити та отримувати дані в режимі реального часу, як у полі, так і в офісі. Сумісність системи з мобільними пристроями особливо корисна для фермерів, які працюють у районах з обмеженим підключенням до Інтернету, оскільки вона підтримує введення даних офлайн з подальшою синхронізацією після встановлення з'єднання.

У модулі молочного тваринництва FarmWizard полегшує запис важливих даних, таких як отелення, бики та лікування. Він бездоганно інтегрується із зовнішніми організаціями, такими як BCMS, агентства з тестування молока та племінні товариства, що забезпечує автоматичне оновлення та зменшує адміністративне навантаження. Крім того, система підтримує моніторинг показників фертильності, дозволяючи фермерам відстежувати такі показники, як коефіцієнти зачаття та показники подання заявок, які є вирішальними для оптимізації продуктивності стада.

Для фермерів, що вирощують м'ясо, FarmWizard надає інструменти для моніторингу як фізичних, так і фінансових показників великої рогатої худоби. Користувачі можуть фіксувати закупівельні ціни, витрати на годівлю та інші пов'язані з ними витрати, пропонуючи уявлення про прибутковість їхньої діяльності. Програмне забезпечення також підтримує вимоги перехресної відповідності та гарантування якості фермерських господарств, гарантуючи, що підприємства з вирощування м'яса дотримуються галузевих стандартів та норм [5].

Модуль FarmWizard для вівчарства спрощує впровадження систем електронної ідентифікації (EID). Він допомагає фермерам фіксувати дані про продуктивність стада, що сприяє прийняттю кращих управлінських рішень та підвищує прибутковість. Система також сприяє веденню точного обов'язкового обліку, забезпечуючи дотримання нормативних вимог та покращуючи відстежуваність.

Інтерфейс введення даних FarmWizard розроблений для ефективності та містить зручні екрани, які дозволяють швидко записувати пакети подій. Система підтримує групування та маркування тварин для швидшої ідентифікації, а також дозволяє кільком користувачам одночасно вводити дані з різних місць. Такий спільний підхід підвищує точність даних і гарантує, що всі члени команди поінформовані та узгоджені.

На завершення, FarmWizard від Orchid FarmTech виділяється як універсальне та орієнтоване на користувача рішення для управління фермерським господарством. Його хмарна інфраструктура, сумісність з мобільними пристроями та комплексні модулі для молочного, м'ясного та вівчарського господарства роблять його цінним інструментом для сучасних тваринницьких операцій. Інтегруючи управління даними, моніторинг ефективності та дотримання нормативних вимог в єдину платформу, FarmWizard надає фермерам можливість приймати обґрунтовані рішення та підвищувати ефективність і прибутковість своїх підприємств.

1.3 Постановка завдання

Основна мета цієї системи полягає у веденні детальних записів даних ідентифікації великої рогатої худоби на основі унікальних дерматогліфічних візерунків, а також відповідної облікової інформації, такої як вік тварини, порода, стан здоров'я та історія переміщень. Для досягнення цієї мети програмне забезпечення розроблено для створення та управління комплексною базою даних, яка зберігає всі необхідні біометричні та адміністративні дані для забезпечення надійної ідентифікації та ефективного відстеження кожної окремої тварини.

Основна функція програмного забезпечення полягає в тому, щоб надати користувачам, таким як фермери, ветеринари та менеджери сільськогосподарських підприємств, швидкий доступ та оновлення інформації, пов'язаної з:

- унікальними дерматогліфічними візерунками кожної тварини для точної ідентифікації;
- історичними даними, включаючи вакцинації, лікування та медичні огляди;
- записами про переміщення та власність для дотримання нормативних вимог;
- статистичними звітами про розмір стада, тенденції здоров'я та показники продуктивності.

Програма працює через зручний інтерфейс з меню, крок за кроком направляючи користувача у введенні нових біометричних даних, оновленні існуючих записів або створенні звітів. Ця діалогова схема налаштування гарантує, що користувачі можуть ефективно орієнтуватися в системі без необхідності спеціалізованих технічних знань.

Крім того, програмне забезпечення надає опції для виправлення або видалення помилкових записів, забезпечуючи точність та актуальність бази даних. Така гнучкість відповідає щоденним операційним потребам і допомагає запобігти невідповідностям даних, які можуть призвести до помилок ідентифікації або проблем із регулюванням [6].

Інтегруючи біометричну ідентифікацію за допомогою дерматогліфів з традиційними функціями обліку великої рогатої худоби, система прагне покращити відстеження, зменшити шахрайство або неправильну ідентифікацію, а також сприяти кращому прийняттю рішень в управлінні худобою.

1.4 Функціональні та нефункціональні вимоги

Функціональні вимоги:

1. збирати, зберігати та обробляти унікальні біометричні візерунки (дерматогліфи) шкіри кожної тварини для точної ідентифікації;

2. створювати, оновлювати та підтримувати детальні профілі для кожної тварини, включаючи вік, породу, стан здоров'я, історію вакцинації та дані про власника;
3. дозволяти уповноваженим користувачам вводити нові біометричні дані, редагувати існуючі записи та видаляти застарілу або неправильну інформацію;
4. забезпечувати швидкий пошук окремих записів про велику рогату худобу за біометричними даними, ідентифікаційними номерами або іншими ідентифікаційними ознаками;
5. реєструвати переміщення тварин між місцями розташування або власниками, щоб забезпечити відстеження та дотримання правил;
6. створювати налаштовувані звіти про демографічні показники стада, тенденції здоров'я та точність ідентифікації для управління фермою та нормативної звітності;
7. надавати доступ на основі ролей, щоб лише уповноважений персонал міг додавати, змінювати або переглядати конфіденційні дані;
8. надсилати сповіщення або нагадування, пов'язані з перевітками здоров'я, вакцинацією або аномаліями, виявленими в біометричних даних;
9. впроваджувати автоматичні механізми резервного копіювання та відновлення, щоб запобігти втраті даних та підтримувати цілісність бази даних.

Нефункціональні вимоги:

1. пропонувати інтуїтивно зрозумілий та простий у навігації інтерфейс користувача, що підходить для користувачів з різними технічними навичками, мінімізуючи вимоги до навчання;

2. оперативно обробляти біометричні дані та запити, щоб забезпечити відповідь у режимі реального або майже реального часу, забезпечуючи безперебійну роботу користувачів;
3. підтримувати високий час безвідмовної роботи та стабільну роботу, використовуючи механізми для обробки системних збоїв та забезпечення безперервної доступності даних;
4. підтримувати зростаючу кількість тварин та користувачів без зниження продуктивності чи швидкості реагування системи;
5. захищати конфіденційні біометричні та персональні дані за допомогою шифрування, безпечної автентифікації та контролю доступу на основі ролей, щоб запобігти несанкціонованому доступу або порушенням;
6. забезпечувати точність та узгодженість даних протягом усіх операцій, включаючи введення даних, оновлення та резервне копіювання.

1.5 Вимоги до інтерфейсу користувача

Інтерфейс користувача програмного забезпечення розроблений для забезпечення безперебійного та простого використання для всіх типів користувачів, включаючи фермерів, ветеринарів та керівників ферм. Він робить акцент на інтуїтивно зрозумілій навігації, що дозволяє користувачам легко переходити між ключовими розділами, такими як профілі великої рогатої худоби, введення біометричних даних, звітність та налаштування системи. Ця чітка структура зменшує плутанину та допомагає користувачам швидко знаходити необхідні інструменти.

Адаптивність є ключовим аспектом інтерфейсу, який створений для безперебійного функціонування на різних пристроях — від настільних комп'ютерів до планшетів та смартфонів. Така оперативність забезпечує

користувачам доступ до даних та керування ними незалежно від того, чи перебувають вони в офісі, чи працюють безпосередньо на фермі [7].

Чітке представлення інформації є пріоритетом. Профілі великої рогатої худоби, дерматогліфічні зображення та медичні записи відображаються за допомогою чітких макетів з продуманою типографікою та кольоровими схемами. Ці дизайнерські рішення покращують читабельність та дозволяють користувачам одразу сприймати важливі деталі.

Процес введення біометричних даних максимально спрощено завдяки добре розробленим формам та інтерактивним інструментам, які допомагають користувачам у фіксації та завантаженні дерматогліфічних візерунків. Корисні підказки та підказки на всіх етапах введення мінімізують помилки та підвищують точність.

Надійні можливості пошуку дозволяють користувачам знаходити певних тварин або групи, застосовуючи фільтри, такі як ідентифікаційні номери, порода, вік або стан здоров'я. Ця функція пришвидшує пошук даних та підтримує ефективне управління стадом.

Інтерфейс забезпечує негайний та чіткий зворотний зв'язок щодо дій користувача, включаючи успішне збереження, оновлення або видалення, а також попередження про виникнення помилок. Така оперативність допомагає користувачам зберігати впевненість та точність під час роботи з системою [7].

Ролі користувачів впливають на зовнішній вигляд та функціональність інтерфейсу, адаптуючи видимі опції до обов'язків кожної людини. Обмежуючи доступ до конфіденційних функцій для неавторизованих користувачів, система забезпечує безпеку та зберігає інтерфейс упорядкованим.

Звітність підтримується через спеціальний розділ, який дозволяє користувачам налаштовувати параметри та створювати детальні звіти.

Зрештою, інтегровані довідкові ресурси та документація легко доступні в системі, що надає інструкції, що дозволяють користувачам повною мірою використовувати можливості програмного забезпечення без значного зовнішнього навчання.

2 МОДЕЛЮВАННЯ ПРЕДМЕТНОЇ ОБЛАСТІ

2.1 Загальні відомості

Уніфікована мова моделювання (UML) слугує стандартизованим способом візуалізації, специфікації, побудови та документування компонентів програмних систем. Це критично важливий інструмент у процесі розробки програмного забезпечення, який дозволяє розробникам та зацікавленим сторонам чітко розуміти архітектуру та поведінку системи ще до початку впровадження [8].

У контексті цього програмного забезпечення для обліку та ідентифікації великої рогатої худоби діаграми UML забезпечують ефективний спосіб ілюстрації структури та взаємодії системи. Різні типи діаграм UML використовуються для охоплення різних перспектив, включаючи статичні та динамічні представлення.

Діаграми класів утворюють основу структурного моделювання системи, деталізуючи класи системи, їх атрибути, методи та зв'язки між ними. У цьому проєкті діаграми класів визначають основні сутності, такі як тварини, біометричні дані, користувачі та звіти, допомагаючи забезпечити відповідність моделей даних реальним вимогам.

Діаграми варіантів використання відображають функціональні вимоги, зображуючи учасників, таких як фермери, ветеринари та адміністратори, та їхню взаємодію з системою. Ці діаграми уточнюють цілі користувачів та реакції системи, допомагаючи визначити область застосування та межі застосування.

Діаграми послідовностей зосереджені на динамічній поведінці системи, ілюструючи потік повідомлень та операцій між об'єктами з часом. Наприклад, процес реєстрації дерматогліфічного візерунка нової тварини або створення звіту про здоров'я можна змоделювати для виявлення необхідних взаємодій та потенційних вузьких місць.

Діаграми діяльності доповнюють діаграми послідовностей, чітко та покроково описуючи робочі процеси та процеси. Вони допомагають візуалізувати логіку складних операцій, таких як збір біометричних даних, перевірка та оновлення записів, покращуючи розуміння процедурних кроків.

Діаграми розгортання надають уявлення про фізичну архітектуру системи, показуючи, як програмні компоненти розподілені між апаратними вузлами. Це особливо важливо для забезпечення належної інтеграції та масштабованості інтерфейсів веб-застосунку, бази даних та клієнта [9].

Разом ці діаграми UML сприяють спілкуванню між розробниками, користувачами та іншими зацікавленими сторонами, зменшують неоднозначності та сприяють добре організованому та зручному програмному рішенню, адаптованому до потреб ідентифікації великої рогатої худоби за допомогою дерматогліфів.

2.2 Об'єктне та функціональне моделювання

2.2.1 Діаграма прецедентів. Діаграма варіантів використання – це візуальне представлення, яке відображає функціональні вимоги до програмної системи, ілюструючи взаємодію між користувачами (акторами) та самою системою. Вона надає загальний огляд того, що система повинна робити з точки зору користувачів, зосереджуючись на їхніх цілях та тому, як система підтримує їх досягнення [10].

Для системи обліку та ідентифікації великої рогатої худоби діаграма варіантів використання визначає основних учасників, таких як фермери, ветеринари та системні адміністратори, кожен з яких має окремі ролі та обов'язки. Ці актори взаємодіють із системою для виконання важливих завдань, таких як реєстрація тварин, збір та управління дерматогліфічними даними, відстеження медичних записів та створення звітів.

Діаграма виділяє ключові варіанти використання, включаючи ідентифікацію тварин, введення біометричних даних, управління записами,

пошук та отримання, звітність та адміністрування користувачів. Відображаючи ці варіанти використання, діаграма уточнює сферу застосування системи, показуючи, які функції доступні кожному типу користувача, та гарантуючи, що всі необхідні функції враховані.

Діаграми варіантів використання також сприяють комунікації між зацікавленими сторонами, надаючи просте та інтуїтивно зрозуміле візуальне уявлення про можливості системи без заглиблення в технічні деталі. Це допомагає у ранній перевірці вимог на етапі розробки та керує наступними етапами проєктування та впровадження.

Підсумовуючи, діаграма варіантів використання виступає як базовий інструмент для узгодження очікувань, визначення меж та організації функціональних вимог системи навколо реальної взаємодії користувачів у контексті ідентифікації великої рогатої худоби за дерматогліфами.

Розроблена діаграма прецедентів використання представлена на рис. 2.1.

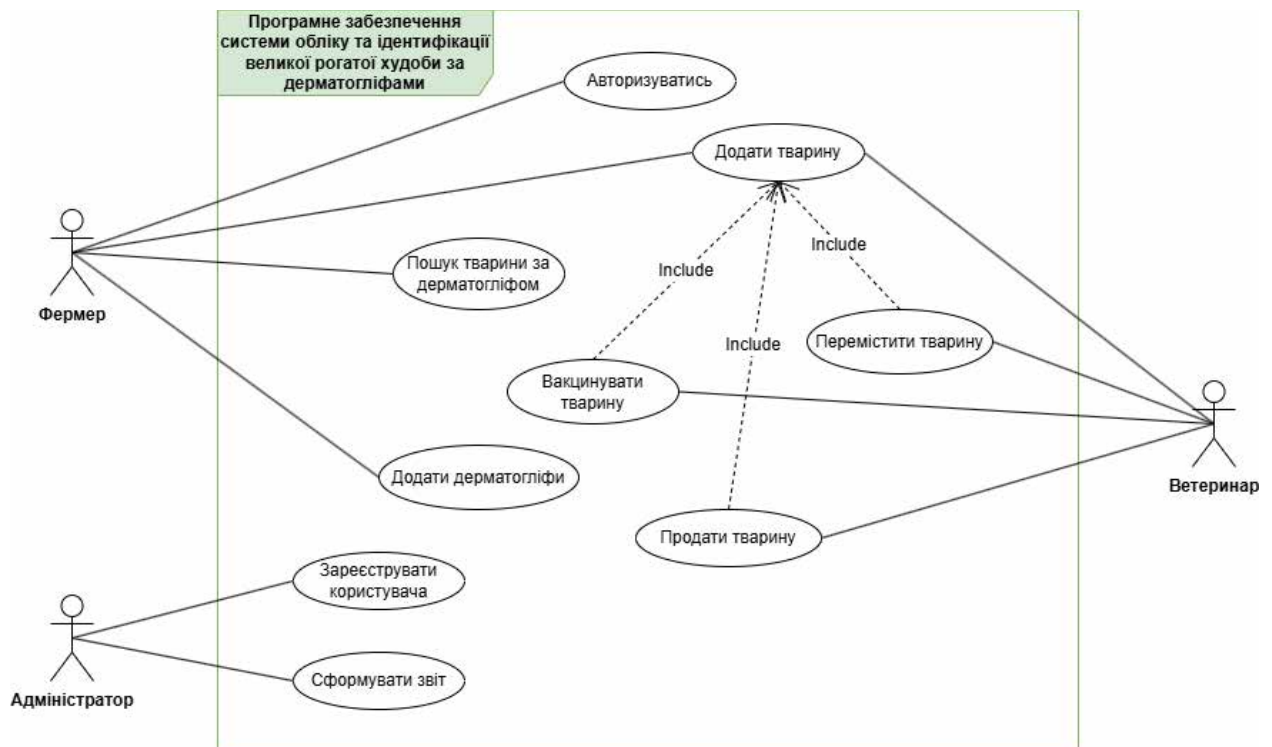


Рис. 2.1 Діаграма прецедентів

Створена діаграма прецедентів містить акторів:

- “Фермер”;
- “Ветеринар”;
- “Адміністратор”.

Актор «Фермер» включає такі прецеденти:

- авторизуватись;
- додати тварину;
- пошук тварини за дерматогліфом;
- додати дерматогліфи.

Актор «Ветеринар» включає такі прецеденти:

- додати тварину;
- перемістити тварину;
- вакцинувати тварину;
- продати тварину.

Актор «Адміністратор» включає такі прецеденти:

- зареєструвати користувача;
- сформувати звіт.

Прецеденти певним чином залежать одне від одного.

Розглянемо детальніше вищеописані прецеденти.

Варіант використання: Додавання дерматогліфа

Актор: Фермер

Опис: Цей варіант використання описує процес, за допомогою якого фермер записує новий дерматогліфічний візерунок для певної тварини в системі. Він включає збір біометричних даних, пов'язування їх з профілем тварини та збереження для подальшої ідентифікації та відстеження.

Основний потік:

1. фермер входить до системи та переходить до розділу управління тваринами;
2. фермер вибирає конкретний профіль тварини, до якого буде додано дерматогліф;
3. фермер ініціює функцію «Додати дерматогліф»;

4. система пропонує фермеру зняти або завантажити зображення чи скан дерматогліфа;
5. фермер надає біометричні дані, завантажуючи файл зображення або використовуючи підключений сканер;
6. система перевіряє якість та повноту введених даних;
7. після успішної перевірки система пов'язує дерматогліф з профілем тварини та зберігає дані;
8. система підтверджує, що дерматогліф успішно додано;
9. фермер виходить з системи або продовжує додавати додаткові біометричні дані за потреби.

Альтернативні потоки:

1. якщо завантажене зображення або скан дерматогліфа нечітке або неповне, система повідомляє фермера та запитує нове захоплення або завантаження. Фермер повторює кроки 4-6, доки дані не будуть прийняті;
2. якщо фермер вибирає недійсний або неіснуючий профіль тварини, система відображає повідомлення про помилку та пропонує вибрати дійсну тварину, перш ніж продовжити;
3. у разі системного збою під час збереження даних з'являється повідомлення про помилку, і фермеру рекомендується повторити операцію або звернутися до служби технічної підтримки.

Випадок використання завершується, коли дерматогліфічні дані успішно записані та пов'язані з вибраним профілем тварини, що дозволяє надійно ідентифікувати їх та використовувати в майбутньому. Цей процес забезпечує ведення точних біометричних записів, підтримуючи загальну мету системи – точний облік та ідентифікацію худоби.

Розглянемо ще один сценарій використання.

Варіант використання: Вакцинація тварини

Актор: Ветеринар

Опис: Цей приклад використання описує процес, за допомогою якого ветеринар реєструє подію вакцинації для конкретної тварини в системі обліку та ідентифікації великої рогатої худоби. Це забезпечує точне ведення записів про вакцинацію як частини профілю здоров'я тварини.

Основний потік:

1. ветеринар входить до системи, використовуючи захищені облікові дані;
2. ветеринар отримує доступ до списку зареєстрованих тварин і вибирає конкретну тварину для вакцинації;
3. ветеринар відкриває розділ записів здоров'я тварини;
4. ветеринар ініціює функцію «Додати запис про вакцинацію»;
5. система відображає форму для введення інформації про вакцинацію, включаючи тип вакцини, номер партії, дату вакцинації та будь-які додаткові примітки;
6. ветеринар точно заповнює інформацію про вакцинацію та надсилає форму;
7. система перевіряє введені дані на повноту та правильність;
8. після успішної перевірки запис про вакцинацію зберігається та пов'язується з профілем здоров'я тварини;
9. система підтверджує успішне додавання запису про вакцинацію;
10. ветеринар може додати додаткові записи про здоров'я або вийти з системи.

Альтернативні потоки:

1. якщо обов'язкові поля відсутні або містять недійсну інформацію (наприклад, неправильний формат дати), система попереджає ветеринара та запитує виправлення, перш ніж продовжити;
2. якщо вибраний профіль тварини не існує або був видалений, система повідомляє ветеринара та пропонує вибрати дійсну тварину;

3. у разі невдачі збереження запису про вакцинацію через системні помилки ветеринар отримує повідомлення про помилку та може повторити операцію або звернутися до служби технічної підтримки.

Випадок використання завершується після успішного запису даних про вакцинацію та їх пов'язування з профілем здоров'я тварини. Це забезпечує надійне відстеження імунізацій, що є важливим для підтримки здоров'я тварин та дотримання ветеринарних стандартів у системі обліку великої рогатої худоби.

2.2.2 Діаграма послідовності. Діаграма послідовностей – це тип UML-діаграми, яка моделює динамічну поведінку системи, показуючи, як об'єкти взаємодіють у певній послідовності для виконання конкретного завдання. Вона зосереджена на порядку повідомлень, що обмінюються між різними компонентами системи або учасниками з часом, що робить її особливо корисною для розуміння та документування робочих процесів або процесів у програмі [11].

У контексті системи обліку та ідентифікації великої рогатої худоби діаграми послідовностей допомагають візуалізувати такі взаємодії, як додавання нового запису дерматогліфів, реєстрація тварини або оновлення інформації про здоров'я. Вони демонструють, як фермер, ветеринар або адміністратор взаємодіє з інтерфейсом системи, як програма обробляє запити та як база даних зберігає або отримує дані.

Деталізуючи покроковий потік комунікації – починаючи від дії користувача, через обробку бізнес-логіки до збереження даних – діаграми послідовностей уточнюють обов'язки кожного компонента системи та розкривають точну послідовність операцій, необхідних для завершення сценарію використання. Це допомагає розробникам ефективніше

проектувати та впроваджувати систему, а також допомагає виявляти потенційні проблеми або вузькі місця на ранніх етапах процесу розробки.

Крім того, діаграми послідовностей сприяють ефективній комунікації між зацікавленими сторонами проєкту, забезпечуючи інтуїтивно зрозуміле візуальне представлення поведінки системи, усуваючи розрив між технічними та нетехнічними членами команди.

Загалом, включення діаграм послідовностей на етапі проєктування програмного забезпечення для ідентифікації великої рогатої худоби забезпечує надійну та зрозумілу модель взаємодії, яка є основою надійної та зручної в обслуговуванні функціональності системи.

Діаграма послідовності, зображена на рис. 2.2, включає наступні об'єкти:

- “Фермер”;
- “Худоба”;
- “Дерматогліф”;
- “Подія”.

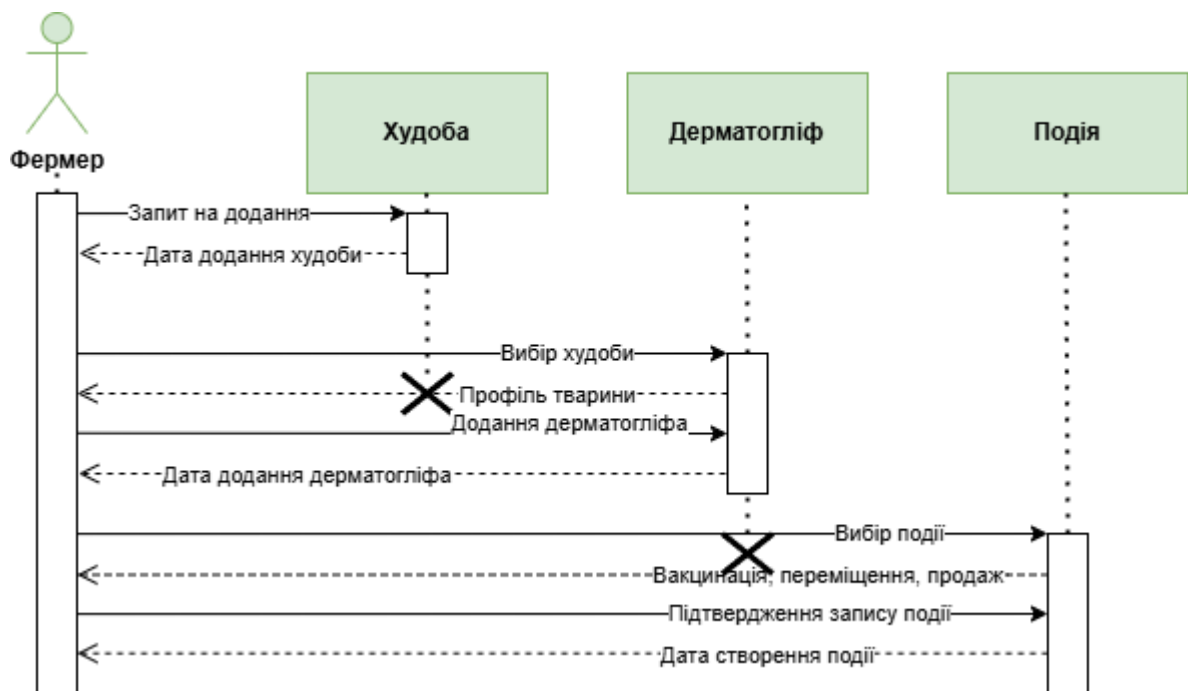


Рис. 2.2 Діаграма послідовності

Діаграма послідовності ілюструє процес, за допомогою якого фермер керує обліком худоби, додаючи біометричні дані та реєструючи події.

Спочатку фермер ініціює введення даних про худобу, вибираючи конкретну тварину та переглядаючи її профіль. Потім система пов'язує унікальний дерматогліф з вибраною худобою, забезпечуючи точну ідентифікацію. Після цього фермер визначає наступний план дій, такий як вакцинація, переміщення або продаж, визначаючи майбутню подію для тварини. Нарешті, система безпечно зберігає цю інформацію, фіксуючи подію разом з датою її створення. Така структурована взаємодія між фермером і системою забезпечує оптимізоване ведення обліку, забезпечуючи точність відстеження діяльності худоби.

2.2.3 Діаграма активності. Діаграма активності – це тип поведінкової діаграми UML, яка візуально представляє потік дій або робочих процесів у системі. Вона підкреслює послідовність та умови координації складних процесів, показуючи, як завдання або дії виконуються від початку до кінця [12].

У контексті вашого програмного забезпечення для обліку та ідентифікації великої рогатої худоби діаграми дій допомагають детально відобразити бізнес-процеси, такі як реєстрація нової тварини, додавання дерматогліфічних даних або реєстрація події вакцинації. Ці діаграми ілюструють покрокові дії, точки прийняття рішень, паралельні завдання та загальний потік, забезпечуючи чітке розуміння функціональності системи.

Виділяючи послідовність дій користувача, реакції системи та умовне розгалуження, діаграми дій допомагають як розробникам, так і зацікавленим сторонам в аналізі та оптимізації робочих процесів. Вони допомагають виявити потенційні вузькі місця або надмірності та

забезпечити відповідність процесів потребам користувачів та системним вимогам.

Діаграми дій також служать комунікаційним мостом, що полегшує технічним та нетехнічним членам команди візуалізацію та узгодження того, як система повинна поводитися під час різних операцій.

Включення діаграм діяльності на етапі проєктування сприяє побудові цілісної та ефективної програмної системи, допомагаючи забезпечити, щоб кожен процес у вашій програмі ідентифікації великої рогатої худоби був добре структурованим, простим для відстеження та належним чином впровадженим.

Розроблена діаграма активності представлена на рис. 2.3

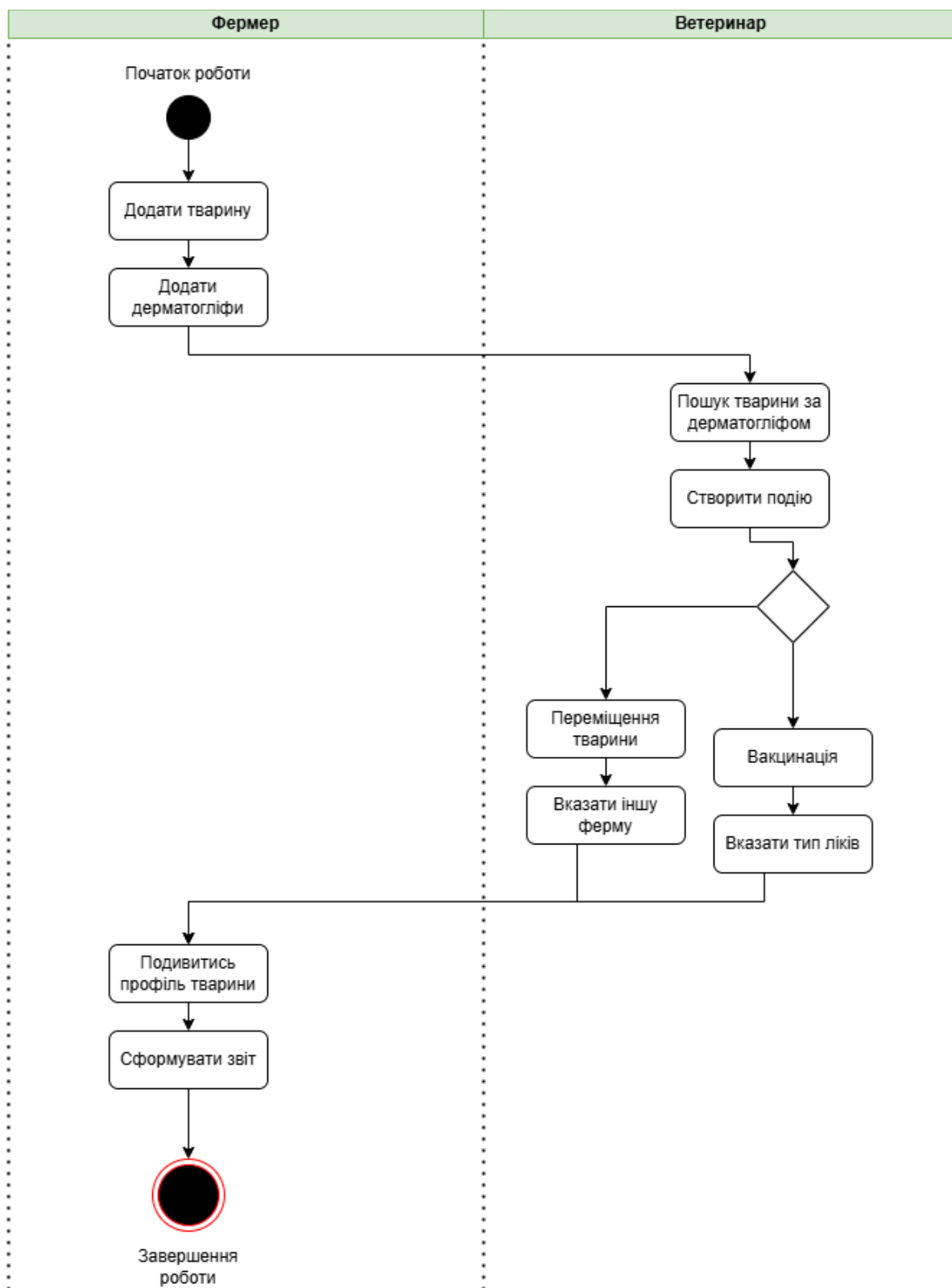


Рис. 2.3 Діаграма активності

Ця діаграма активності демонструє послідовність дій, які фермер виконує при додаванні худоби, дерматогліфа та події. Вона відображає основні взаємодії між фермером та системою управління худобою.

Процес розпочинається із запиту на додавання худоби, після чого фермер обирає потрібну тварину та переглядає її профіль. Наступний етап включає внесення дерматогліфа, що є унікальним біометричним параметром для ідентифікації худоби. Після цього фермер визначає подію, яка має бути зареєстрована—це може бути вакцинація, переміщення або продаж тварини. Завершальним етапом є підтвердження запису події, при якому фіксується дата її створення.

Діаграма чітко демонструє логіку роботи системи, забезпечуючи точність у веденні записів та спрощуючи управління худобою.

2.3 Абстракції предметної області

Розробка програмної системи для ідентифікації та обліку великої рогатої худоби на основі дерматогліфів вимагає продуманої абстракції реального світу в керовані програмні компоненти. Ці абстракції слугують основою для моделювання сутностей, їх атрибутів та взаємозв'язків у цифровому середовищі.

В основі системи лежить сутність «Велика рогата худоба», яка представляє окремих тварин у стаді. Кожна тварина абстрагується за допомогою таких атрибутів, як ідентифікаційний номер, порода, вік, стать, стан здоров'я та, найголовніше, дані дерматогліфів – унікальний ідентифікатор на основі шаблону, подібний до відбитків пальців людини. Цей дерматогліф діє як біометричний підпис, забезпечуючи надійну та захищену від несанкціонованого втручання ідентифікацію.

Ще однією важливою абстракцією є запис дерматогліфів, який інкапсулює дані біометричного сканування або зображення, метадані, такі як дата та метод збору, а також будь-які маркери перевірки або контролю якості. Ці записи безпосередньо пов'язані з конкретними тваринами, що забезпечує послідовне відстеження [13].

Користувачі системи поділяються на такі ролі, як фермери, ветеринари та адміністратори. Кожна роль має різні дозволи та схеми взаємодії з системою. Фермери відповідають за реєстрацію тварин та дерматогліфів, тоді як ветеринари можуть оновлювати дані про здоров'я або історію вакцинації. Адміністратори контролюють цілісність даних та доступ користувачів.

Система також абстрагує події або дії, такі як вакцинація, медичні огляди та передача права власності. Ці події пов'язані з обліком худоби та мають часову позначку, створюючи хронологічну історію життєвого циклу кожної тварини.

Ці сутності підтримує системний інтерфейс, абстракція рівня взаємодії користувача. Він розроблений інтуїтивно зрозумілим та специфічним для певної ролі, що дозволяє кожній групі користувачів ефективно виконувати завдання — чи то завантаження зображень, перегляд профілів худоби, чи створення звітів.

Ці абстракції разом утворюють концептуальну модель предметної області, гарантуючи, що програмне забезпечення відображає складність та специфіку управління худобою, залишаючись при цьому організованим, масштабованим та зручним для користувача.

Абстракція: Худоба	Абстракція: Подія
Властивості: Номер Порода Дата створення Дерматогліфи Характеристики	Властивості: ПІБ виконавця Худоба Тип події Дата події Опис
Обов'язки: Пошук за дерматогліфом Додати худобу Створити подію Додати до ферми	Обов'язки: Перемістити худобу Вакцинувати худобу Продати худобу

Рис. 2.4 Абстракції предметної області

В основі предметної області управління худобою лежать кілька ключових абстракцій, які допомагають формалізувати процеси та забезпечити точний облік.

Одна з основних абстракцій — худоба. Вона включає всі характеристики тварини, такі як порода, вік, стан здоров'я та унікальні біометричні дані. Ця абстракція дозволяє вести детальні записи про кожну одиницю худоби, забезпечуючи її ідентифікацію та контроль.

Абстракція події відіграє центральну роль у системі, оскільки всі важливі дії щодо худоби записуються у вигляді подій. Сюди входять вакцинація, переміщення між локаціями, продаж або інші значущі зміни. Кожна подія має свою дату створення та деталізацію, що робить систему управління худобою більш прозорою та ефективною.

Усі ці абстракції працюють у взаємозв'язку, формуючи цілісну модель управління худобою, де кожен елемент відіграє свою роль у підтримці точності та контролю в системі.

2.4 Діаграма класів

Діаграма класів — це центральна структурна діаграма в UML (Уніфікована мова моделювання), яка представляє статичну структуру системи. Вона визначає ключові класи, їхні атрибути, операції (методи) та зв'язки між ними. У контексті вашої системи обліку великої рогатої худоби та ідентифікації дерматогліфів діаграма класів служить планом, який відображає, як програмне забезпечення буде організовано з точки зору об'єктів та взаємодії даних.

В основі системи лежить клас *Animal* (Тварина), який містить важливу інформацію про кожну окрему корову чи бика, таку як її унікальний ідентифікатор, порода, стать, дата народження та власність. Цей клас тісно пов'язаний з класом *Dermatoglyph* (Дерматогліф), який містить дані, пов'язані з біометричним шаблоном, що використовується для ідентифікації, зазвичай

зберігаються у вигляді зображення, разом з датою захоплення та оцінкою якості.

Клас User (Користувач) розроблений для підтримки різних ролей у системі, включаючи фермерів, ветеринарів та адміністраторів. Він містить загальні дані користувача, такі як ім'я, контактні дані, облікові дані автентифікації та дозволи на основі ролей. Залежно від ролі, користувач може мати доступ до різних операцій у системі.

Клас VaccinationRecord пов'язує конкретних тварин з даними про вакцинацію, фіксуючи такі деталі, як тип вакцини, ветеринар, який її проводив (пов'язаний з класом User), та дату введення. Це забезпечує повне медичне відстеження стада.

Для підтримки взаємодії система також включає такі класи, як LoginSession для керування автентифікованими сесіями користувачів, Report для створення зведень та документації, а також AuditLog для відстеження дій користувачів та змін у системі для підзвітності та простежуваності.

Асоціації, такі як зв'язки «один до багатьох» або «багато до багатьох», візуалізуються між класами, щоб показати, як сутності даних взаємопов'язані. Наприклад, один користувач (фермер) може володіти багатьма тваринами, і кожна тварина може мати кілька записів Dermatoglyph та VaccinationRecord.

Ця діаграма класів виступає основою для розробки програмного забезпечення, забезпечуючи цілісність, зручність підтримки та масштабованість об'єктно-орієнтованої структури, одночасно точно моделюючи процес ідентифікації великої рогатої худоби за допомогою дерматогліфів [15].

Для цього проєкту було створено спеціальну діаграму класів з використанням об'єктно-орієнтованого підходу (рис. 2.5).

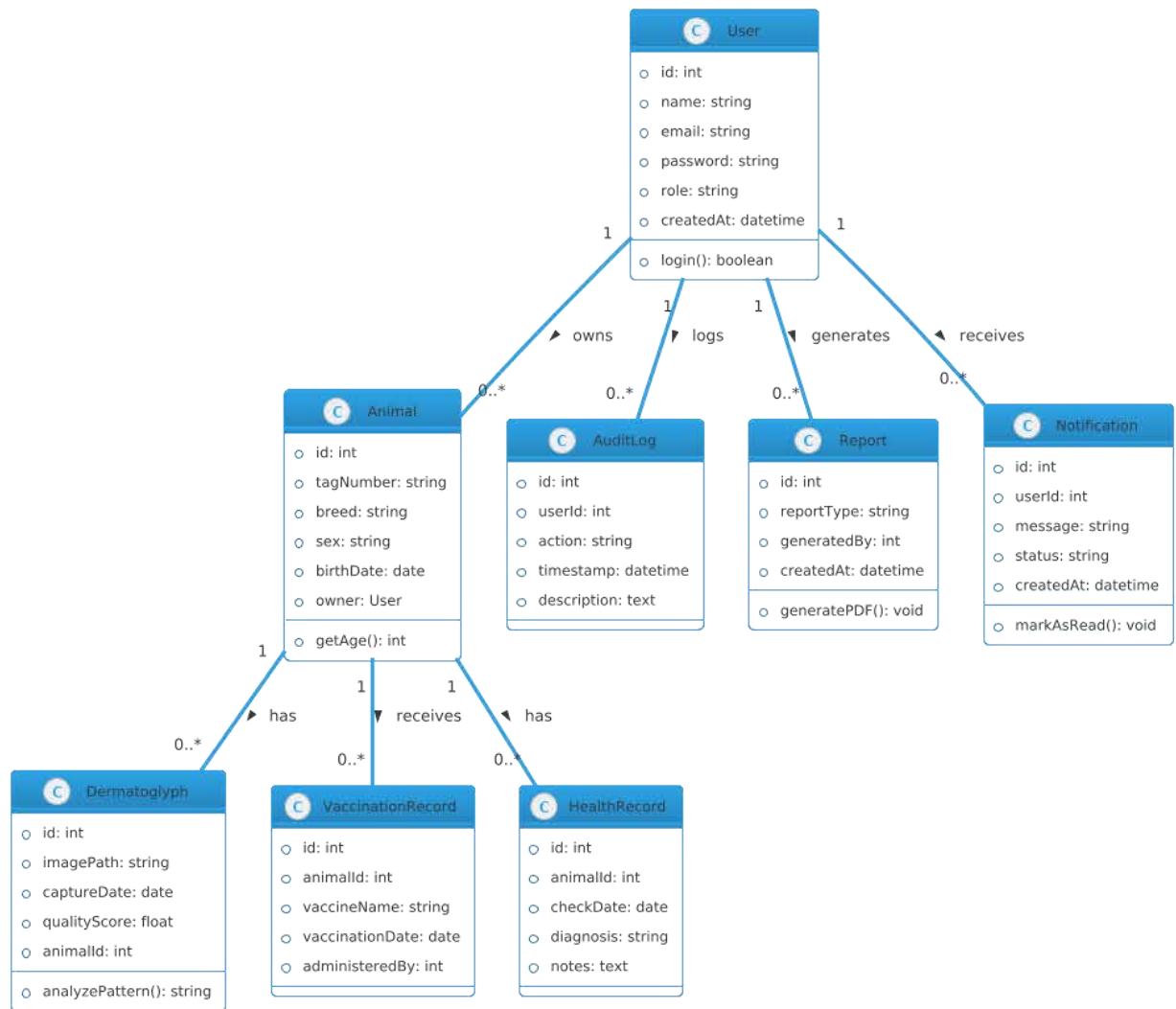


Рис. 2.5 Діаграма класів

Ця діаграма класів відображає взаємодію між фермером та основними об'єктами системи: худоба, дерматогліф, подія. Вона показує структуру і зв'язки між класами, демонструючи логіку управління записами.

Фермер має можливість додавати худобу, переглядати її профіль, а також асоціювати дерматогліф—унікальний біометричний параметр, який забезпечує точну ідентифікацію тварини. Крім цього, фермер може створювати події, такі як вакцинація, переміщення чи продаж, зберігаючи їх разом із датою створення.

Ця структура дозволяє ефективно вести записи та контролювати кожну дію, пов'язану з худобою, забезпечуючи точний облік і управління інформацією.

3 ПРОЄКТУВАННЯ ПРОГРАМНОЇ СИСТЕМИ

3.1 Логічна модель даних

Логічна модель даних (LDM) представляє абстрактну структуру даних системи, зосереджуючись на сутностях, їх атрибутах та зв'язках, незалежно від будь-якої технології фізичної бази даних. Ця модель служить мостом між бізнес-вимогами та реалізацією бази даних, чітко ілюструючи, як інформація в системі ідентифікації великої рогатої худоби організована, пов'язана та логічно зберігається [16].

У центрі моделі знаходиться сутність **Animal** (Тварина), яка містить такі атрибути, як **AnimalID** (Ідентифікатор тварини), **TagNumber** (Номер мітки), **Breed** (Порода), **Sex** (Стать) та **Birthdate** (Дата народження). Кожна тварина унікально ідентифікується та відстежується в системі. Ця сутність має зв'язок «один до багатьох» з сутністю **Dermatoglyph** (Дерматогліф), яка містить біометричні дані, такі як **ImagePath** (Шлях до зображення), **CaptureDate** (Дата захоплення) та **QualityScore** (Кваліфікаційний показник). Кожен запис дерматогліфа пов'язаний з певною твариною та служить основним методом біометричної ідентифікації.

Сутність **User** (Користувач) фіксує учасників системи, включаючи фермерів, ветеринарів та адміністраторів. Вона містить такі поля, як **UserID** (Ідентифікатор користувача), **Name** (Ім'я), **Email** (Електронна пошта), **PasswordHash** (Хеш пароля) та **Role** (Роль). Зв'язки встановлюються для відображення обов'язків — наприклад, користувач (фермер) може володіти кількома тваринами, а користувач (ветеринар) може бути пов'язаний з кількома записами про вакцинацію.

Для підтримки відстеження здоров'я модель включає сутності **VaccinationRecord** та **HealthRecord**. Обидві пов'язані з сутністю **Animal** та зберігають важливі дані, такі як **VaccineName** (Назва вакцини),

VaccinationDate (Дата вакцинації), Diagnosis (Діагноз) та Notes (Примітки). Ці записи підтримують відстеження та історію хвороби кожної тварини.

Підтримуючи підзвітність системи, сутність AuditLog (Журнал аудиту) веде хронологічний запис дій користувачів, зберігаючи такі деталі, як UserID (Ідентифікатор користувача), ActionType (Тип дії), Timestamp (Позначка часу) та Description (Опис). Ця сутність є важливою для відстеження конфіденційних операцій та забезпечення цілісності системи.

Також включено додаткові сутності, такі як Report (Звіт) та Notification (Сповідження). Reports (Звіти) зберігають зведені дані, згенеровані користувачами, для аналізу та відповідності, тоді як Notifications (Сповідження) інформують користувачів про майбутні події, нагадування або сповіщення.

Кожна сутність нормалізована для мінімізації надмірності та забезпечення узгодженості. Зв'язки між ними – здебільшого один до багатьох – чітко визначені за допомогою зовнішніх ключів, які згодом перетворюються на фактичні обмеження у фізичній базі даних.

Логічна модель даних – це важливий інструмент планування, який гарантує, що всі бізнес-правила, робочі процеси та залежності даних фіксуються у структурованому та логічному форматі, готуючи основу для майбутнього проєктування схеми бази даних та впровадження системи.

Логічна модель системи представлена на рисунку 3.1.

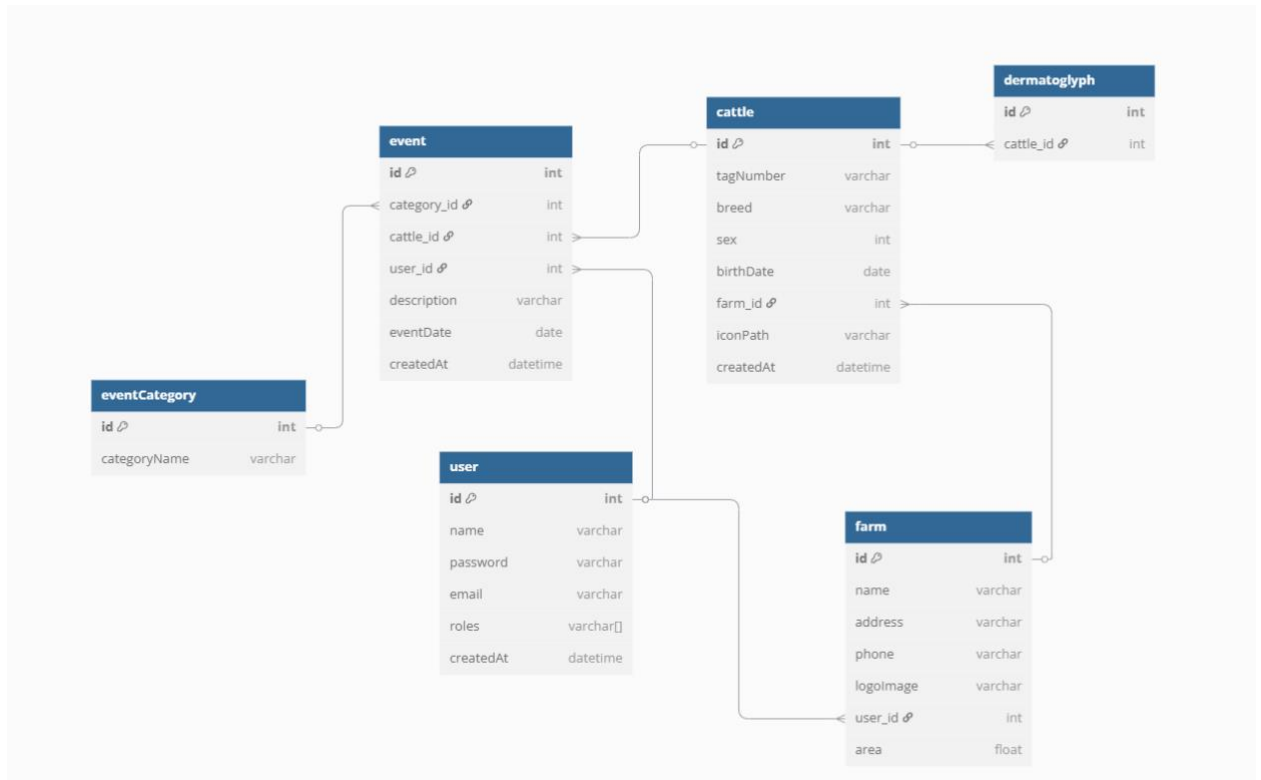


Рис. 3.1 ER-діаграма

Таблиці та їхні поля

User

1. id (int, primary key, auto-increment): Унікальний ідентифікатор для кожного користувача;
2. name (varchar): Ім'я користувача;
3. password (varchar): Хешований пароль користувача;
4. email (varchar): Адреса електронної пошти користувача;
5. roles (varchar array): Ролі користувачів, наприклад, admin, звичайний користувач;
6. createdAt (datetime): Мітка часу створення користувача.

Farm

1. id (int, primary key, auto-increment): Унікальний ідентифікатор для кожної ферми;
2. name (varchar): Назва ферми;
3. address (varchar): Адреса ферми;
4. phone (varchar, nullable): Контактний номер телефону ферми;

5. `logoImage (varchar, nullable)`: Шлях або дані зображення логотипу ферми;
6. `user_id (int, зовнішній ключ до user.id)`: Власник або користувач, пов'язаний з фермою;
7. `area (float, nullable)`: Розмір/площа ферми.

Cattle

1. `id (int, primary key, auto-increment)`: Унікальний ідентифікатор для кожної худоби;
2. `tagNumber (varchar)`: Ідентифікаційний номер бирки для худоби;
3. `breed (varchar, nullable)`: Порода худоби;
4. `sex (int)`: Стать худоби (ймовірно, перерахунок або код);
5. `birthDate (date, nullable)`: Дата народження худоби;
6. `farm_id (int, зовнішній ключ до farm.id)`: Ферма, якій належить худоба;
7. `iconPath (varchar)`: Шлях до значка/зображення, що представляє худобу;
8. `createdAt (datetime)`: Коли було створено запис про худобу.

Dermatoglyph

1. `id (int, primary key, auto-increment)`: Унікальний ідентифікатор для кожного запису дерматогліфа;
2. `cattle_id (int, зовнішній ключ до cattle.id)`: Худоба, до якої належить цей дерматогліф.

EventCategory

1. `id (int, primary key, auto-increment)`: Унікальний ідентифікатор для кожної категорії подій;
2. `categoryName (varchar)`: Назва категорії події (наприклад, "Вакцинація", "Медичний огляд").

Event

1. `id (int, первинний ключ, автоінкремент)`: Унікальний ідентифікатор для кожної події;

2. `category_id` (int, зовнішній ключ до `eventCategory.id`): Категорія події;
3. `cattle_id` (int, зовнішній ключ до `cattle.id`): Худоба, що брала участь у події;
4. `user_id` (int, зовнішній ключ до `user.id`): Користувач (ймовірно, персонал або ветеринар), відповідальний за подію;
5. `description` (varchar, nullable): Опис або примітки щодо події;
6. `eventDate` (дата): Дата, коли відбулася подія;
7. `createdAt` (дата та час): Коли було створено запис події.

У цій моделі даних кожна ферма пов'язана з одним користувачем, який виступає її власником. Цей зв'язок встановлюється полем `user_id` ферми, що вказує на унікальний ідентифікатор користувача. Аналогічно, велика рогата худоба пов'язана з фермами, тобто кожна тварина належить до однієї конкретної ферми, пов'язаної через `farm_id` худоби, що посилається на ідентифікатор ферми.

Записи дерматогліфів, які зберігають біометричні дані, пов'язані з великою рогатою худобою, безпосередньо пов'язані з окремою худобою. Кожен запис дерматогліфа посилається на відповідну худобу через її `cattle_id`. Події, які документують випадки або дії, пов'язані з великою рогатою худобою, мають кілька зв'язків. Кожній події присвоюється категорія, яка пов'язує її з `eventCategory`, і цей зв'язок підтримується через поле `category_id` події. Події також пов'язані з великою рогатою худобою через `cattle_id`, що вказує, якої тварини стосується подія.

Крім того, події пов'язані з користувачами, які представляють осіб, відповідальних за створення або управління цими записами, причому поле `user_id` встановлює цей зв'язок. Це створює комплексну структуру, де користувачі володіють фермами, ферми утримують кілька голів худоби, і кожна худоба може мати кілька біометричних записів та пов'язаних з ними подій. Самі події організовані за категоріями та пов'язані з користувачами, які їх контролюють. Ця взаємопов'язана структура підтримує ефективне

управління худобою, веденням медичних записів та пов'язаною з ними діяльністю.

3.2 Вибір системи управління базою даних та її реалізація

Критичним компонентом у розробці ефективної програмної системи для обліку та ідентифікації худоби за допомогою дерматогліфів є вибір відповідної системи керування базами даних (СКБД). СКБД повинна бути здатною ефективно обробляти складні реляційні дані, забезпечувати цілісність даних, підтримувати одночасний доступ кількох користувачів та забезпечувати масштабованість у міру зростання системи [17].

Враховуючи ці вимоги, було обрано реляційну систему баз даних завдяки її надійності в управлінні структурованими даними та зв'язками між такими сутностями, як користувачі, ферми, худоба, записи дерматогліфів, події та категорії подій. Обрана СКБД сприяє організації даних у взаємопов'язані таблиці, що дозволяє виконувати ефективні запити та надійні транзакції.

У реалізації використовувалася сучасна платформа об'єктно-реляційного картування (ORM) для поєднання програмного застосунку з базовою базою даних. Цей підхід абстрагує операції з базою даних в керовані об'єкти в середовищі програмування, підвищуючи швидкість розробки та зручність обслуговування, зберігаючи при цьому високу узгодженість даних.

Завдяки впровадженню цієї СКБД система підтримує комплексне управління обліком худоби — пов'язуючи кожну тварину з її фермою, пов'язуючи біометричні дані дерматогліфів та відстежуючи події, пов'язані зі здоров'ям або ідентифікацією. Архітектура забезпечує послідовне зберігання всієї відповідної інформації, що дозволяє здійснювати надійні процеси ідентифікації та обліку, необхідні для ефективного управління худобою.

Рішення використовувати MySQL як систему управління базами даних для програмного забезпечення для обліку та ідентифікації худоби було

прийнято на основі кількох важливих факторів. По-перше, MySQL — це широко поширена, надійна та зріла реляційна система баз даних, відома своєю високою продуктивністю та стабільністю, що є важливим для обробки структурованих даних, пов'язаних з управлінням худобою [18].

Її здатність ефективно керувати складними взаємозв'язками між такими об'єктами, як ферми, велика рогата худоба, записи дерматогліфів та події, робить її природним вибором для цього проєкту. MySQL підтримує розширені механізми запитів та індексації, які забезпечують швидке отримання біометричних даних та пов'язаної з ними інформації, що має вирішальне значення для своєчасної ідентифікації та ведення обліку.

Ще однією ключовою причиною є широка підтримка MySQL спільнотою та вичерпна документація, що сприяє швидшій розробці та усуненню несправностей. Наявність численних інструментів та варіантів інтеграції, включаючи сумісність з популярними фреймворками об'єктно-реляційного картографування (ORM), спрощує процес розробки та підвищує зручність обслуговування.

Крім того, MySQL пропонує масштабованість, що дозволяє системі зростати та обробляти зростаючі обсяги даних у міру додавання нових ферм та записів про велику рогату худобу. Його підтримка цілісності транзакцій забезпечує узгодженість даних, що є критично важливим для підтримки точних та надійних записів ідентифікації.

Підсумовуючи, баланс продуктивності, надійності, простоти використання та масштабованості MySQL зробив його оптимальним вибором для реалізації рівня бази даних системи ідентифікації великої рогатої худоби, забезпечуючи довгострокову ефективність та стійкість проєкту.

Таким чином було створено базу даних в середовищі MySQL Workbench (Рис. 3.2).

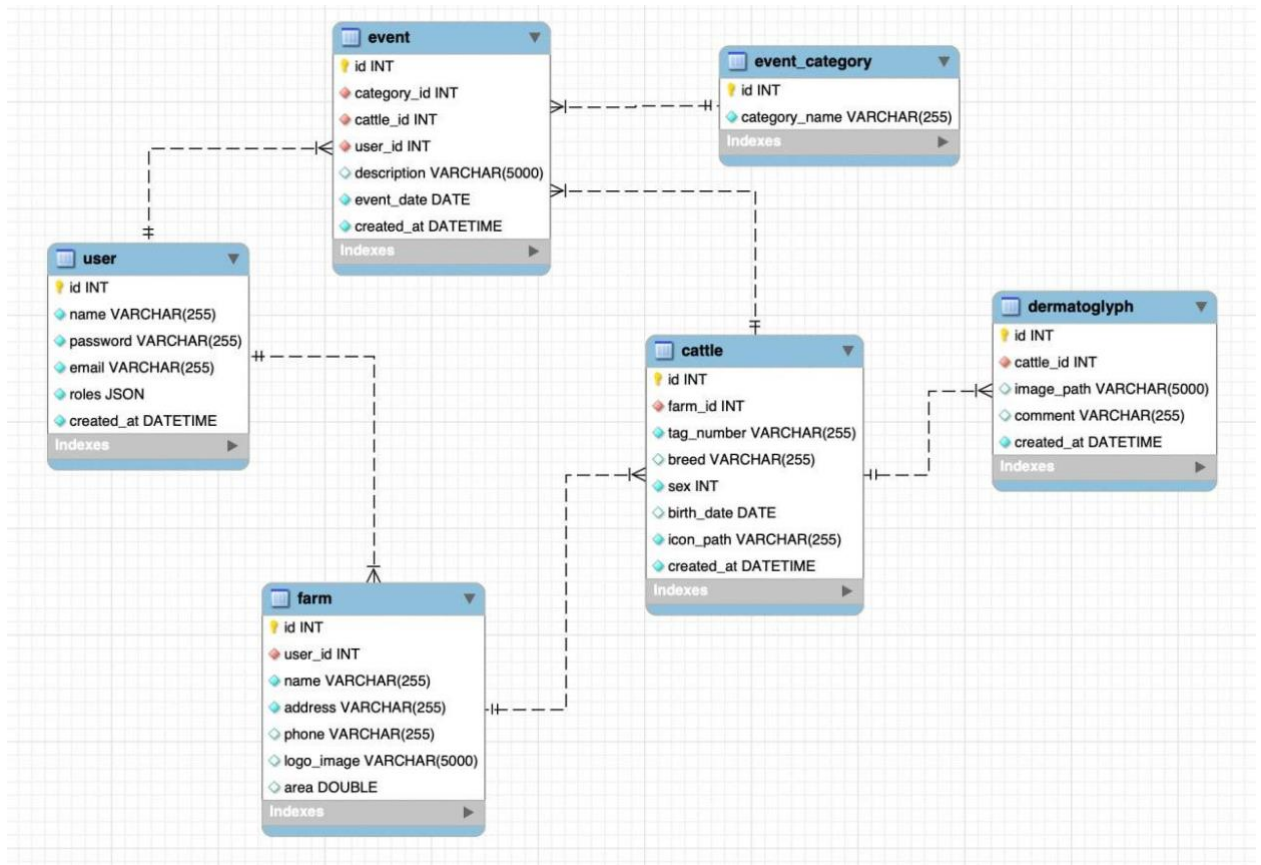


Рис. 3.2 База даних

Ця база даних структурує процеси управління худобою, дерматогліфами та подіями, забезпечуючи точний облік та контроль. Вона включає три основні категорії: худоба, дерматогліф та події, які взаємодіють між собою.

Система дозволяє фермеру додавати нову худобу, фіксуючи її характеристики та дату внесення до бази. Кожній тварині можна призначити дерматогліф — унікальний біометричний параметр, що допомагає ідентифікувати її та запобігати підміні. Крім цього, фермер реєструє події, такі як вакцинація, переміщення або продаж, зберігаючи їх разом із датою створення.

Завдяки цій базі даних фермер отримує надійний механізм для ведення точного обліку худоби, спрощення адміністративного процесу та покращення ефективності управління господарством.

3.3 Архітектура програмного забезпечення

Програмна система для обліку та ідентифікації великої рогатої худоби побудована з використанням багаторівневого архітектурного підходу, який забезпечує модульність, масштабованість та легкість обслуговування. Її дизайн зосереджений на ефективному управлінні даними, пов'язаними з фермами, окремими поголів'ями великої рогатої худоби, біометричними дерматогліфами та пов'язаними з ними подіями, тим самим підтримуючи точну ідентифікацію та ефективне управління ресурсами.

На найвищому рівні рівень представлення виступає інтерфейсом між користувачами та системою. Він забезпечує зручне та адаптивне середовище, адаптоване для різних ролей, включаючи власників ферм, ветеринарів та системних адміністраторів. Цей рівень полегшує введення, відображення та управління даними, обробляючи взаємодію з користувачами та перевірку перед передачею запитів до основної логіки [19].

Під рівнем представлення знаходиться рівень логіки застосунку або бізнес-логіки, який служить серцем системи. Він забезпечує дотримання бізнес-правил, обробляє вхідні дані та керує робочими процесами, такими як ідентифікація великої рогатої худоби та класифікація подій. Централізація цих правил в одному місці гарантує узгодженість та спрощує оновлення або розширення функціональності системи.

Верхні рівні підтримує рівень даних, який відповідає за постійне зберігання та пошук інформації. Реалізований через реляційну систему баз даних, таку як MySQL, цей рівень організовує дані у структуровані сутності, такі як ферми, велика рогата худоба, дерматогліфи, події та користувачі. Він ефективно обробляє складні запити та транзакції, зберігаючи цілісність даних.

Щоб подолати концептуальні відмінності між кодом об'єктно-орієнтованих додатків та структурою реляційної бази даних, система використовує інструмент об'єктно-реляційного відображення (ORM). Цей компонент спрощує маніпулювання даними, абстрагуючи взаємодії з базою

даних, зменшуючи повторюваний код та підвищуючи продуктивність розробників.

Безпека є невід'ємною частиною архітектури, включаючи механізми автентифікації та авторизації, щоб гарантувати захист конфіденційних даних та те, що доступ до інформації або її зміну можуть отримувати лише дозволені користувачі [20].

Дизайн архітектури робить акцент на розширюваності, дозволяючи безперешкодно інтегрувати нові функції, такі як розширена біометрична обробка або комплексна звітність, без порушення роботи існуючих компонентів.

Завдяки цій багаторівневій та модульній архітектурі система підтримує чіткий розподіл обов'язків, полегшуючи обслуговування та забезпечуючи зростання для задоволення потреб управління та ідентифікації великої рогатої худоби, що постійно змінюються.

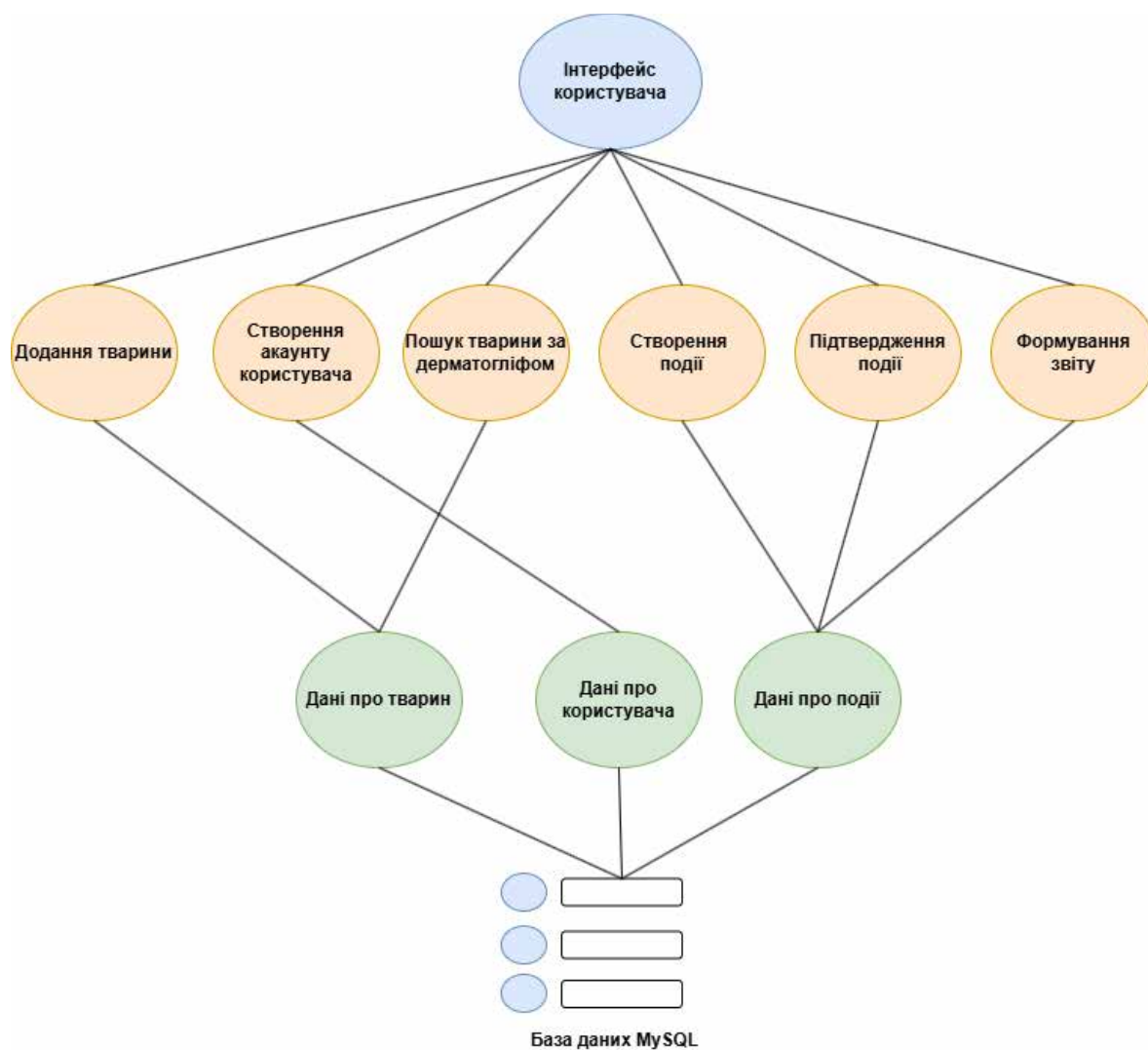


Рис. 3.3 Багатошарова архітектура

Ця діаграма відображає чотиришарову архітектуру програмного забезпечення, що використовується для управління худобою. Вона містить ключові рівні: Фермер, Худоба, Дерматогліф та Подія, кожен з яких виконує свою роль у загальному процесі.

На першому рівні фермер ініціює запит на додавання худоби. Далі система реєструє вибрану тварину, пов'язуючи її з унікальним дерматогліфом—біометричною ознакою, яка гарантує точну ідентифікацію. Наступний шар відповідає за визначення події, що може включати вакцинацію, переміщення або продаж. Завершальним етапом є підтвердження події та її збереження разом із датою створення.

Така структурована архітектура забезпечує ефективний запис та управління інформацією, спрощуючи контроль над худобою і підвищуючи точність обліку.

3.4 Організаційна структура програмного забезпечення

3.4.1 Діаграма пакетів. Діаграма пакетів організовує компоненти системи в логічні групи, що відображають їхні ролі та обов'язки, сприяючи чіткій та керованій структурі. Основні пакети включають керування користувачами, управління фермою, управління худобою, обробку дерматогліфів, відстеження подій та збереження даних.

Ці пакети взаємодіють через чітко визначені інтерфейси, зберігаючи модульність та дозволяючи розробникам незалежно підтримувати або розширювати окремі частини системи. Така структура сприяє чіткому визначенню меж відповідальності, що робить систему легшою для розуміння та розвитку з часом [20].

Діаграма пакета — це тип структурної діаграми, що використовується в програмній інженерії для організації та групування пов'язаних елементів системи в пакети. Вона забезпечує загальний огляд архітектури системи, показуючи, як різні частини системи логічно розділені та як вони залежать одна від одної або взаємодіють одна з одною.

Пакети подібні до контейнерів або модулів, які об'єднують класи, інтерфейси або інші пакети з подібною функціональністю або призначенням. Це допомагає керувати складністю, розбиваючи систему на керовані розділи, що полегшує її розуміння, розробку та підтримку.

Діаграма зазвичай ілюструє такі зв'язки, як залежності, асоціації або імпорт між пакетами. Візуалізуючи ці зв'язки, розробники та зацікавлені сторони можуть побачити загальну структуру системи, зрозуміти, як організовані компоненти, та визначити області зв'язку або модульності.

По суті, діаграма пакета допомагає передати організацію програмної системи на вищому рівні абстракції, підтримуючи кращий дизайн, чіткішу документацію та ефективну співпрацю.

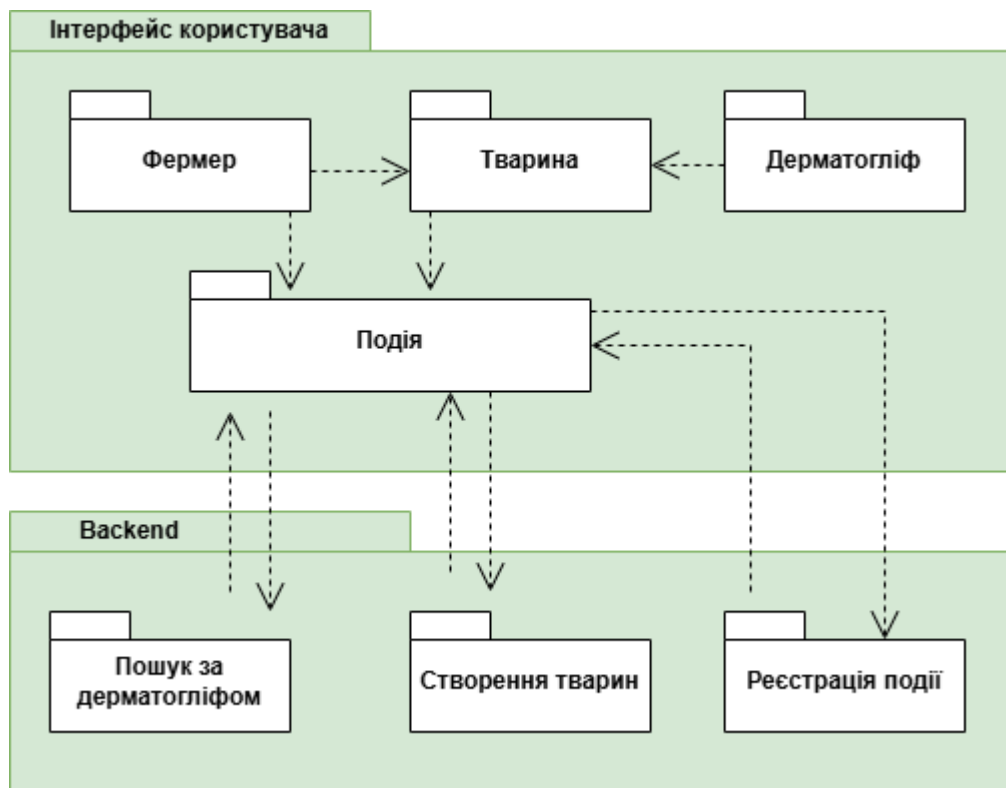


Рис. 3.4 Діаграма пакетів

Ця діаграма пакетів демонструє структуроване розподілення функціональних компонентів у системі управління худобою. Вона відображає логічні взаємозв'язки між основними пакетами, забезпечуючи впорядковане опрацювання даних.

Діаграма включає чотири ключові пакети: Фермер, Худоба, Дерматогліф та Подія. Кожен пакет містить відповідні класи та взаємодіє з іншими для виконання основних процесів. Фермер має доступ до пакета худоби, що дозволяє переглядати профіль тварини, здійснювати реєстрацію та оновлення даних. Пакет дерматоглівів відповідає за біометричну ідентифікацію худоби, забезпечуючи точне розпізнавання кожної одиниці. Події фіксуються у відповідному пакеті, де систематизуються всі значущі дії, такі як вакцинація, переміщення чи продаж.

Такий підхід до структурування системи дозволяє ефективно керувати обліком худоби, мінімізувати ризики втрати даних та забезпечити надійну взаємодію між елементами.

3.5 Вибір інструментарію для створення програмного забезпечення

Для розробки програмної системи, призначеної для обліку та ідентифікації худоби за допомогою дерматогліфів, було ретельно підібрано набір інструментів, щоб забезпечити ефективність, надійність та зручність обслуговування. Розробка серверної частини базується на PHP, широко використовуваній мові серверних сценаріїв, відомій своєю універсальністю та сильною підтримкою спільноти. Для розширення можливостей PHP та оптимізації структури програми використовується фреймворк Symfony. Symfony забезпечує надійну основу з компонентами багаторазового використання та відповідає архітектурі MVC, що забезпечує чисту та модульну організацію коду.

Для управління даними MySQL обрано як систему управління базами даних завдяки його перевірній стабільності, масштабованості та сумісності з PHP. Цей вибір полегшує обробку складних зв'язків даних, властивих записам ідентифікації худоби та відстеженню подій.

Обраним інтегрованим середовищем розробки (IDE) є PhpStorm, яке пропонує потужні інструменти для розробки на PHP та Symfony, включаючи інтелектуальне автодоповнення коду, налагодження та інтеграцію контролю версій. Git використовується для управління вихідним кодом та контролю версій, підтримуючи спільну розробку та відстеження історії коду.

На стороні фронтенду стандартні веб-технології, такі як HTML, CSS та JavaScript, забезпечують структуру, стиль та інтерактивність користувацького інтерфейсу. Для пришвидшення розробки інтерфейсу користувача та забезпечення адаптивності на різних пристроях використовується Bootstrap,

популярний CSS-фреймворк. Таке поєднання гарантує зручний та візуально узгоджений інтерфейс.

Крім технічних переваг, вибір цих інструментів зумовлений їхньою широкою підтримкою в IT-спільноті та наявністю великої кількості навчальних матеріалів, що суттєво полегшує процес розробки та подальшої підтримки системи. Symfony і PHP мають добре документовані API, а також активні форуми, що дозволяє оперативно вирішувати можливі проблеми та інтегрувати нові функції без значних витрат часу.

Окрім цього, використання Git для контролю версій забезпечує не лише збереження історії змін у коді, але й організовує спільну роботу розробників, дозволяючи одночасно працювати над різними модулями системи та легко інтегрувати їх у єдину структуру. Такий підхід підвищує продуктивність команди та знижує ризики виникнення помилок при об'єднанні коду.

Разом ці інструменти утворюють комплексну екосистему розробки, що забезпечує ефективне кодування, надійну взаємодію з базами даних, безперебійну співпрацю та адаптивний користувацький досвід.

4 ВПРОВАДЖЕННЯ ТА ЕКСПЛУАТАЦІЯ СИСТЕМИ

4.1 Вимоги до апаратного та програмного забезпечення

Для успішної розробки та експлуатації програмної системи, призначеної для обліку та ідентифікації великої рогатої худоби за допомогою дерматогліфів, необхідно виконати певні вимоги до апаратного та програмного забезпечення, щоб забезпечити безперебійну роботу та продуктивність.

Що стосується апаратного забезпечення, система потребує надійного серверного середовища, здатного ефективно запускати внутрішні служби. Це включає сучасний багатоядерний процесор з достатньою кількістю оперативної пам'яті (рекомендовано щонайменше 8 ГБ) для одночасної обробки даних та запитів користувачів без затримки. Обсяг пам'яті має бути достатнім та швидким, бажано використовувати SSD-накопичувачі, для підтримки операцій з базою даних та зберігання файлів зображень дерматогліфів. Для цілей розробки та тестування достатньо персональних комп'ютерів з аналогічними характеристиками. Крім того, робочі станції користувачів потребують стандартних периферійних пристроїв, таких як монітори, клавіатури, миші та, можливо, сканери або камери високої роздільної здатності для захоплення візерунків дерматогліфів у великої рогатої худоби.

Щодо програмних вимог, система побудована на PHP-бекенді з використанням фреймворку Symfony, що вимагає сумісного веб-сервера, такого як Apache або Nginx, з PHP 8 або вище. Система керування базами даних MySQL повинна бути встановлена та налаштована для обробки зберігання даних та запитів. Для взаємодії з фронтендом необхідні сучасні

веббраузери, що підтримують HTML5, CSS3 та JavaScript, щоб належним чином отримати доступ до інтерфейсу користувача системи.

Інструменти розробки включають PHPStorm як основне середовище розробки (IDE) для ефективного кодування, налагодження та управління проєктами, а також Git для контролю версій для управління спільною розробкою. Система також залежить від різних бібліотек та фреймворків, таких як Bootstrap для адаптивного веб-дизайну.

Разом ці вимоги до апаратного та програмного забезпечення забезпечують основу для розробки, розгортання та підтримки надійної, ефективної та зручної системи, яка відповідає конкретним потребам ідентифікації великої рогатої худоби за допомогою дерматогліфіки.

Також було зроблено діаграму розгортання для візуального огляду деплою системи (Рис. 4.1).

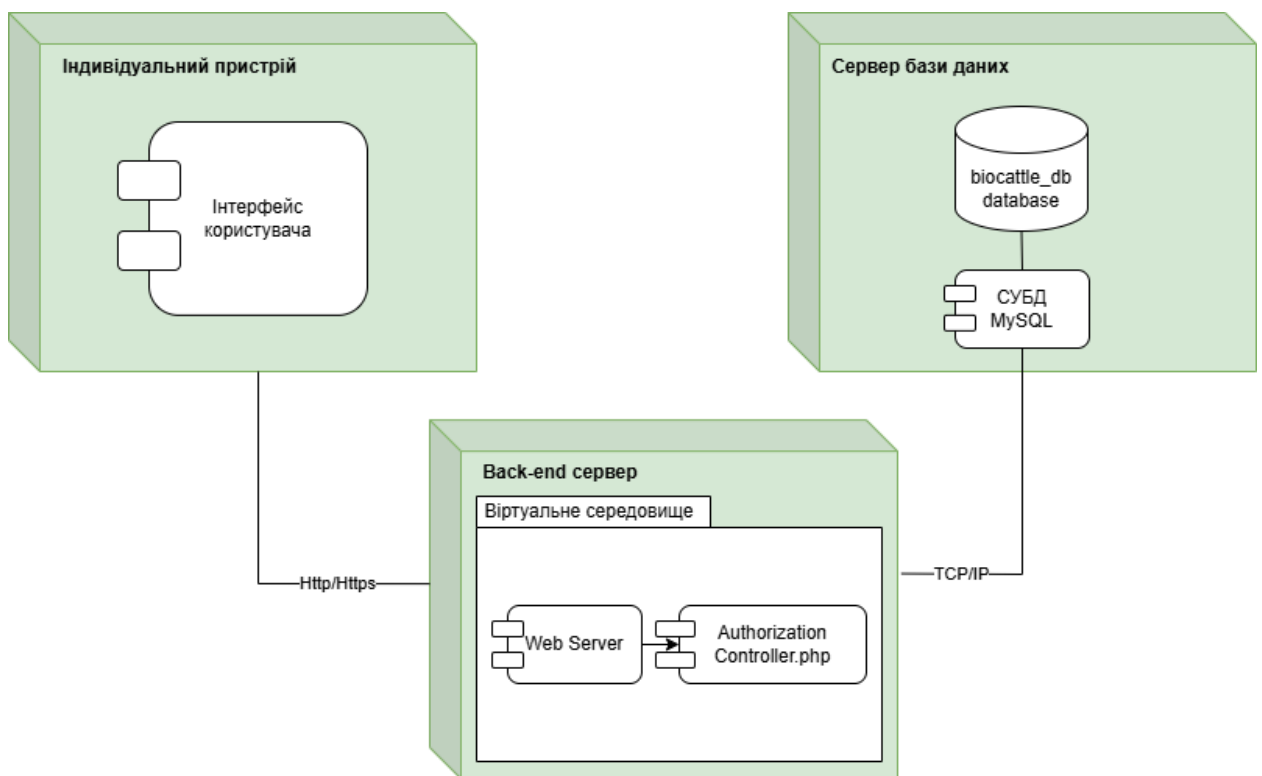


Рис. 4.1 Діаграма розгортання

4.2 Тестування системи

Запустивши систему, потрапляємо на форму авторизації, тут нам треба ввести логін та пароль користувача (рис. 4.2).

Рис. 4.2 Форма авторизації

Після авторизації ми переходимо на головну сторінку програми. Тут у нас виводиться різна статистична інформація про облік тварин та дерматогліфи, а також особисті виконані події (Рис. 4.3).

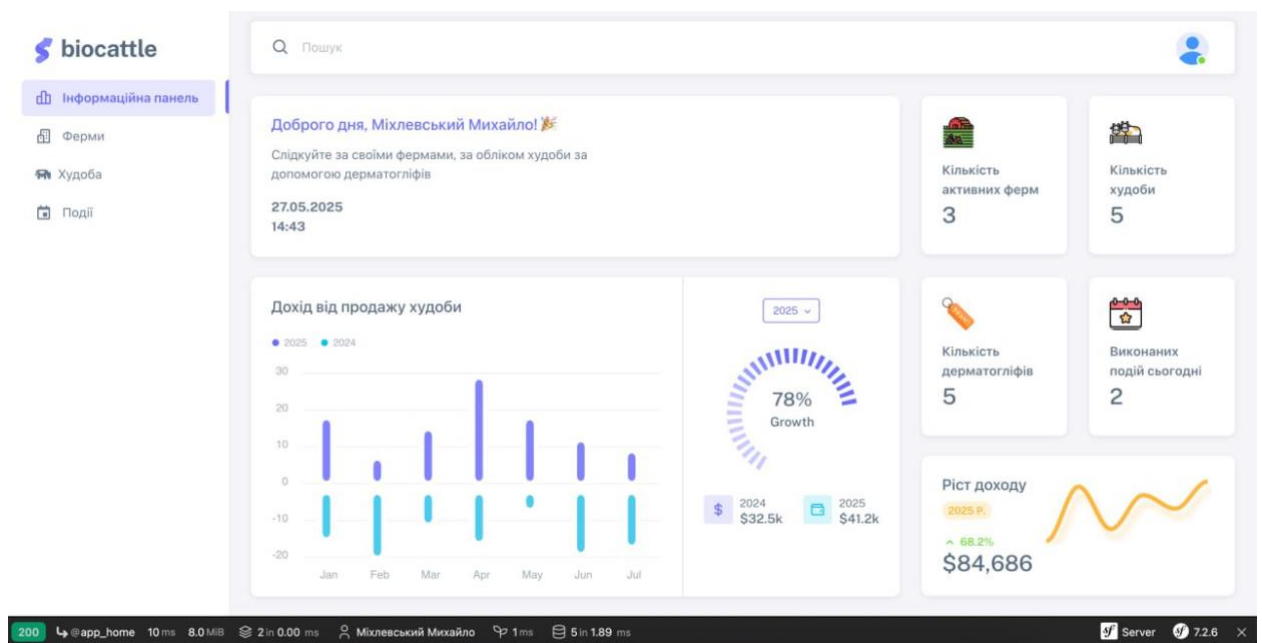


Рис. 4.3 Головна сторінка

Зайдемо на сторінку "Ферми", щоб побачити список створених нами ферм нашого підприємства. У кожній фермі є тварина, яка була прив'язана до ферми під час її додавання в систему (Рис 4.4).

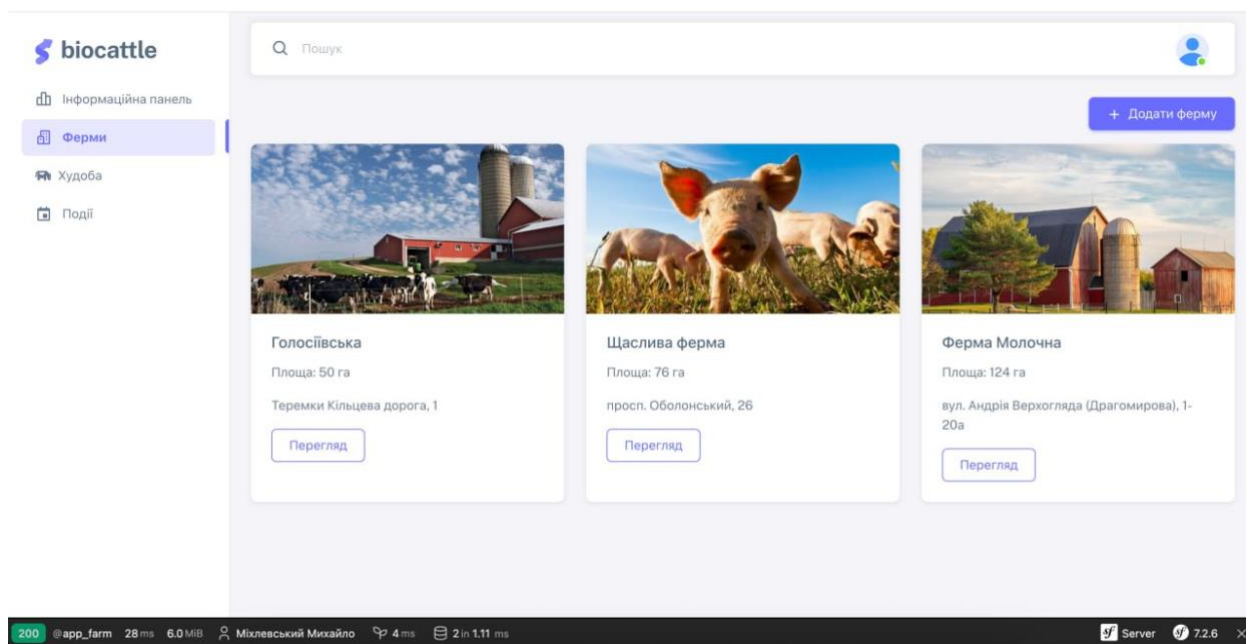


Рис. 4.4 Сторінка "Ферми"

Щоб додати ферму, натискаємо на кнопку "Додати ферму" і у нас відкриється сторінка з полями для введення всієї необхідної інформації, яку нам потрібно ввести (Рис. 4.5)

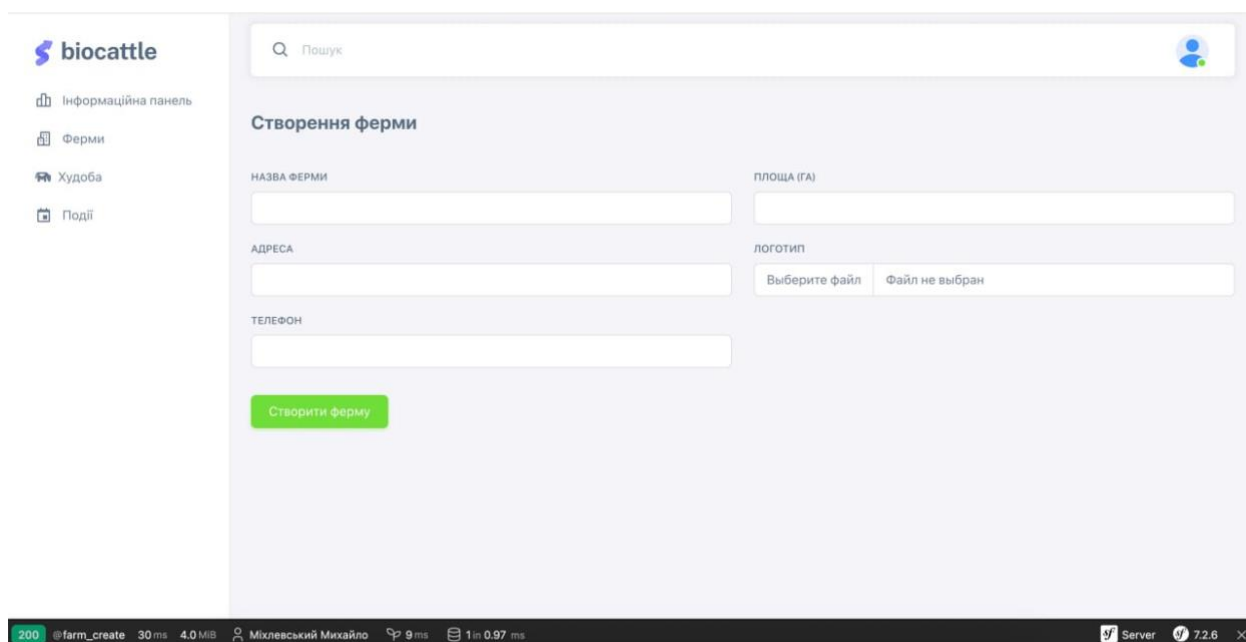


Рис. 4.5 Форма створення ферми

Після цього можемо детально подивитися на додану ферму і побачити всю інформацію про неї, а також про тварин, які числяться на ній (Рис. 4.6).

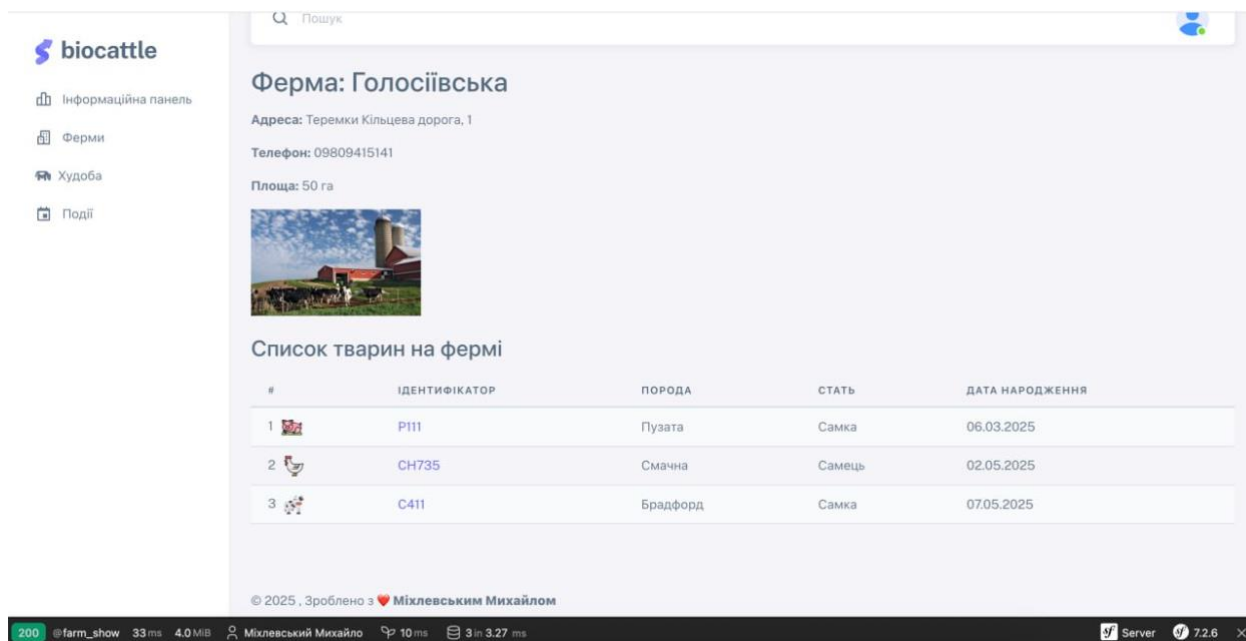


Рис. 4.6 Огляд ферми

Далі переходимо на сторінку "Худоба". Тут у нас виводиться список усіх зареєстрованих тварин у системі (Рис. 4.7).

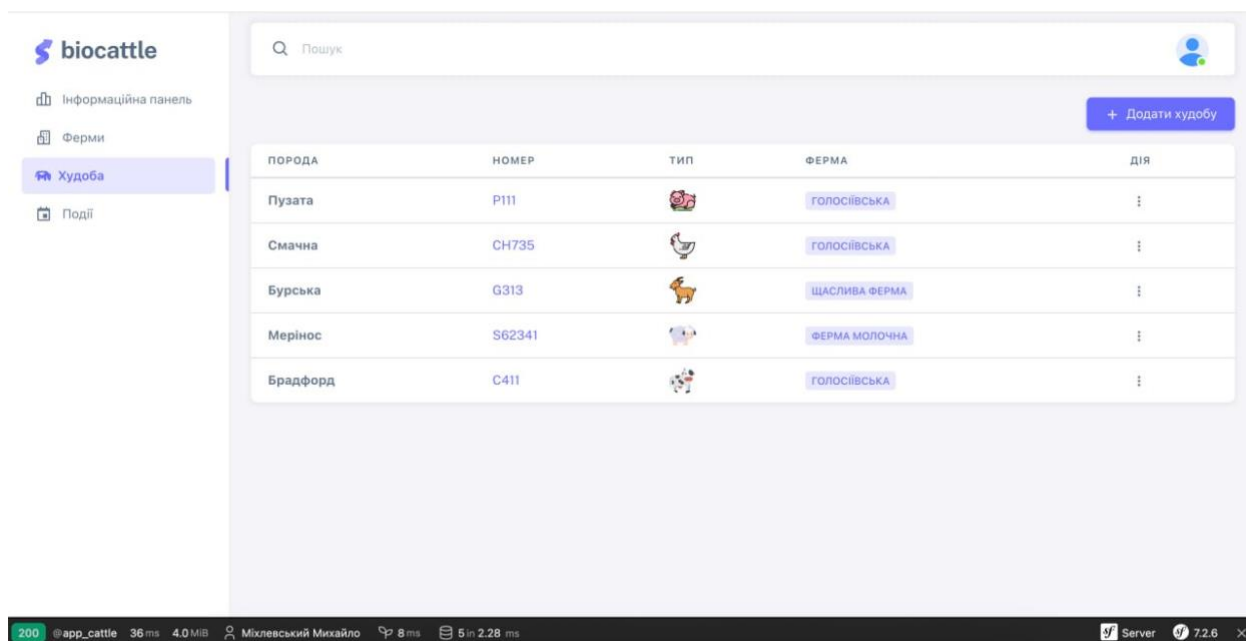
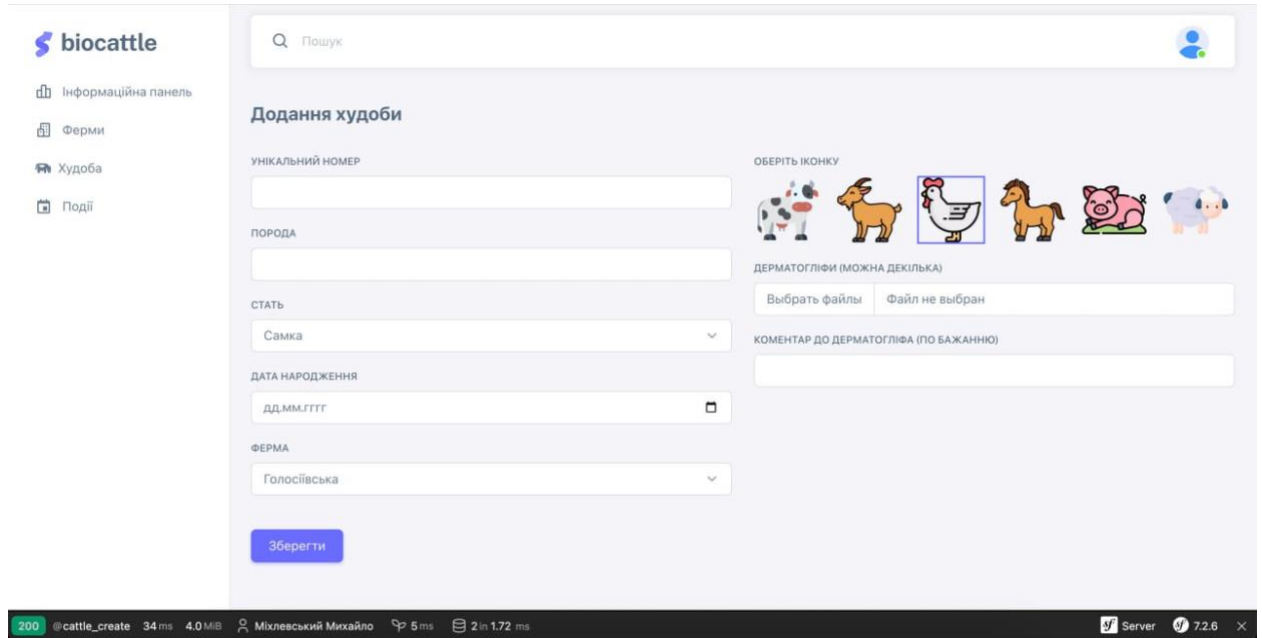


Рис. 4.7 Сторінка "Худоба"

Спробуємо зареєструвати нову тваринку, для цього натискаємо кнопку "Додати худобу", яка відкриває нам сторінку для заповнення даних про тварину, тут нам потрібно вказати всі дані, вибрати іконку тварини та завантажити дерматогліфи у вигляді фотографій (Рис. 4.8).



The screenshot displays the 'biocattle' web application interface. On the left is a sidebar with navigation links: 'Інформаційна панель', 'Ферми', 'Худоба', and 'Події'. The main content area is titled 'Додання худоби' (Add Animal). It contains several input fields: 'Унікальний номер' (Unique number), 'Порода' (Breed), 'Стать' (Sex) with a dropdown menu showing 'Самка' (Female), 'Дата народження' (Date of birth) with a date picker, and 'Ферма' (Farm) with a dropdown menu showing 'Голосіївська'. To the right of these fields is a section for selecting an animal icon, labeled 'ОБЕРІТЬ ІКОНКУ', with icons for a cow, goat, chicken, horse, pig, and sheep. Below the icons is a section for uploading dermographs, labeled 'ДЕРМАТОГЛІФИ (МОЖНА ДЕКІЛЬКА)' (Dermographs (you can upload several)). It includes a file picker with buttons 'Вибрати файли' (Select files) and 'Файл не вибрано' (File not selected), and a text area for 'КОМЕНТАР ДО ДЕРМАТОГЛІФА (ПО БАЖАННЮ)' (Comment on dermograph (if desired)). A blue 'Зберегти' (Save) button is at the bottom left of the form. The browser's developer console is visible at the bottom, showing network requests and performance metrics.

Рис. 4.8 Форма додання тварини

Так само можемо подивитися на дерматогліф який ми додали до нашої тварини, для цього нам треба натиснути на номер бирки обраної тварини (Рис. 4.9).

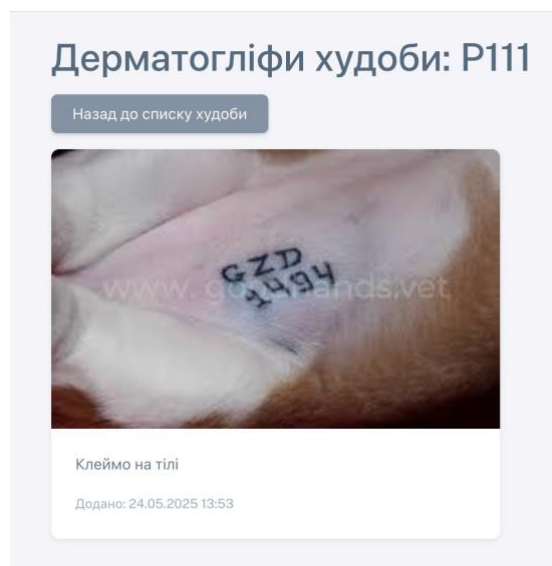
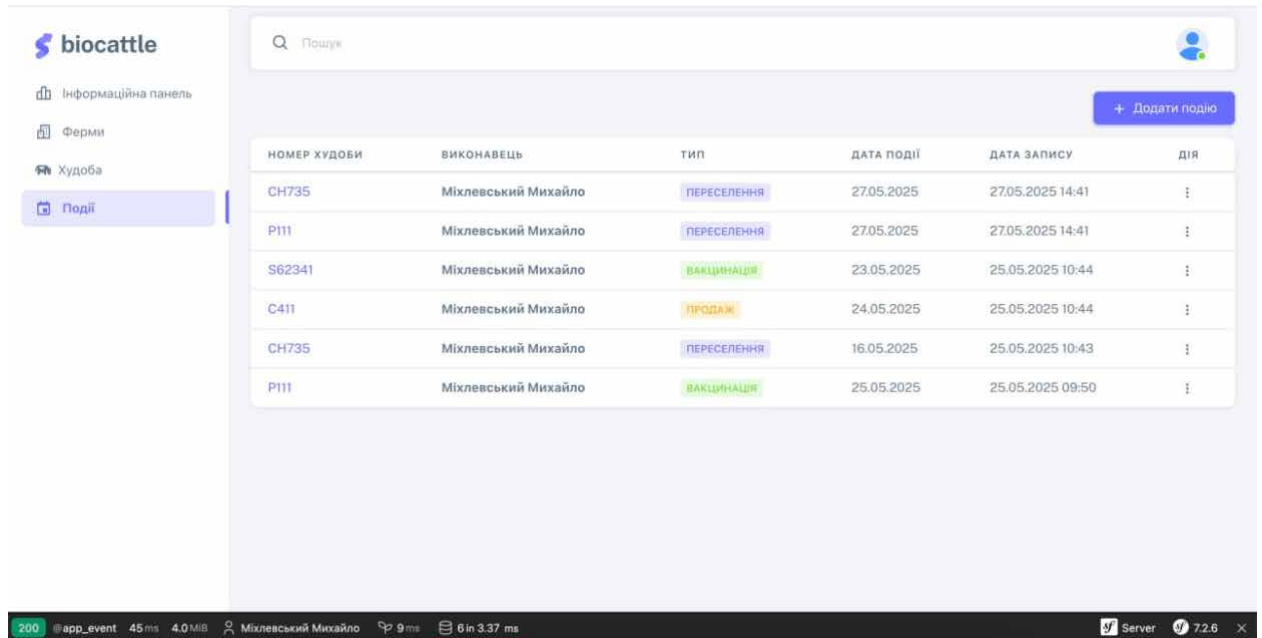


Рис. 4.9 Перегляд дерматогліфу тварини

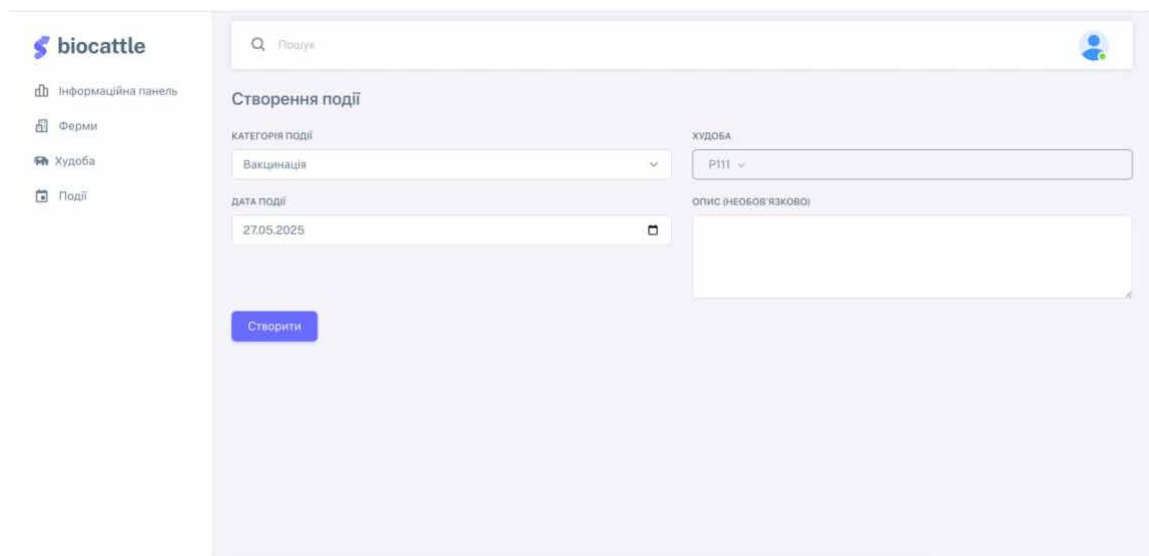
Переходимо на сторінку "Події". Тут ми маємо список всіх останніх подій на підприємстві. Події поділяються на переселення, вакцинацію та продаж. У події завжди вказується працівник та тварина (Рис. 4.10).



НОМЕР ХУДОБИ	ВИКОНАВЕЦЬ	ТИП	ДАТА ПОДІЇ	ДАТА ЗАПИСУ	ДІЯ
CH735	Міхлевський Михайло	ПЕРЕСЕЛЕННЯ	27.05.2025	27.05.2025 14:41	⋮
P111	Міхлевський Михайло	ПЕРЕСЕЛЕННЯ	27.05.2025	27.05.2025 14:41	⋮
S62341	Міхлевський Михайло	ВАКЦИНАЦІЯ	23.05.2025	25.05.2025 10:44	⋮
C411	Міхлевський Михайло	ПРОДАЖ	24.05.2025	25.05.2025 10:44	⋮
CH735	Міхлевський Михайло	ПЕРЕСЕЛЕННЯ	16.05.2025	25.05.2025 10:43	⋮
P111	Міхлевський Михайло	ВАКЦИНАЦІЯ	25.05.2025	25.05.2025 09:50	⋮

Рис. 4.10 Сторінка “Події”

Для того щоб створити подію, натискаємо на відповідну кнопку, перед нами відкривається сторінка створення події, де ми вибираємо її тип, вибираємо тварину зі списку та опис за бажанням, підтверджуємо дію та подію реєструється в системі (Рис. 11).



Створення події

КАТЕГОРІЯ ПОДІЇ:

ХУДОБА:

ДАТА ПОДІЇ:

ОПИС (НЕОБОВ'ЯЗКОВО):

Рис. 4.11 Форма додання події

ВИСНОВКИ

Розробка програмної системи для обліку та ідентифікації худоби за допомогою дерматогліфів відповідає на значну потребу сучасного управління тваринництвом. Традиційні методи ідентифікації, такі як вушні бирки або татуювання, часто стикаються з проблемами, пов'язаними з довговічністю, підробкою та точністю. Використовуючи дерматогліфічні візерунки, які є унікальними та незмінними, система забезпечує високонадійний біометричний підхід до ідентифікації окремих тварин. Це вдосконалення гарантує, що кожну корову можна чітко розпізнати протягом усього її життєвого циклу, що значно зменшує кількість помилок та покращує відстеження.

У процесі розробки вибір інструментів та технологій був критично важливим для досягнення цілей системи. Поєднання PHP та фреймворку Symfony дозволило швидку та структуровану розробку зручного та масштабованого веб-додатку. MySQL було обрано як систему управління базами даних завдяки його надійності, широкому використанню та здатності ефективно обробляти складні реляційні дані. Разом з фронтенд-технологіями, такими як HTML, CSS, JavaScript та Bootstrap, проєкт призвів до створення зручного інтерфейсу, доступного для персоналу ферм з різним рівнем технічної експертизи.

Архітектура системи була розроблена для інтеграції даних про ферми, худобу, дерматогліфи та пов'язані з ними події узгодженим та організованим чином. Такий цілісний підхід дозволяє користувачам не лише ідентифікувати тварин, але й керувати обліком ферми, відстежувати події здоров'я та розмноження, а також вести повну історію кожної тварини. Поєднуючи біометричні дані з операційними записами, програмне забезпечення підтримує прийняття обґрунтованих рішень та покращує загальну практику управління фермою.

Впровадження цього програмного забезпечення також вимагало уваги до вимог до обладнання та інфраструктури, що гарантувало можливість розгортання системи в реальних сільськогосподарських умовах. Залежність від стандартних налаштувань сервера та поширених програмних інструментів зменшує бар'єри для впровадження та підтримує масштабованість. Крім того, використання контролю версій за допомогою Git та середовища розробки PHPStorm сприяло професійному та спільному робочому процесу, забезпечуючи ефективну розробку, тестування та майбутні оновлення.

Окрім безпосередніх функціональних переваг, цей проєкт підкреслює зростаючу важливість цифрової трансформації в сільському господарстві. Інтеграція біометричних систем ідентифікації, таких як аналіз дерматогліфів, відкриває нові можливості для покращення добробуту тварин, боротьби з шахрайством та оптимізації програм розведення. У міру того, як ферми продовжують модернізуватися, такі технології ставатимуть дедалі ціннішими для відповідності нормативним стандартам та ринковим вимогам щодо прозорості та відстежуваності.

Загалом, завершена система є значним кроком вперед у технології ідентифікації худоби. Він поєднує надійні біометричні методи з практичним розробкою програмного забезпечення, щоб забезпечити комплексний інструмент для управління фермерським господарством. Проєкт демонструє потенціал сучасних програмних фреймворків та баз даних для вирішення реальних сільськогосподарських проблем. Подальша робота може розширити можливості системи за допомогою таких функцій, як мобільний доступ, розширена аналітика та інтеграція з іншими інструментами управління фермерським господарством, що ще більше підвищить її корисність та вплив.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Сміт, Б. П. Внутрішня медицина великих тварин. — Elsevier, 2015.
2. Джайн, А. К., Росс, А., Нандакумар, К. Вступ до біометрії. — Springer, 2016.
3. Вільямс, П., Гопінат, А. Проєктування баз даних та реляційна теорія: нормальні форми та все таке інше. — O'Reilly Media, 2018.
4. AgriWebb — [Електронний ресурс] — Режим доступу: <https://www.agriwebb.com>
5. FarmWizard — [Електронний ресурс] — Режим доступу: <https://www.eyesparcs.com/projects/farm-wizard/>
6. Еванс, Дж., Гарсія, М. Системи ідентифікації та відстеження худоби. — Wiley-Blackwell, 2019.
7. Фаулер, С., Рід, Дж. Освоєння Symfony. — Packt Publishing, 2020.
8. Хан, М. А., Лі, Х. Дж. Технології ідентифікації тварин: принципи та застосування. — CRC Press, 2017.
9. Ульман, Дж. Д. Системи баз даних: повна книга. — Pearson, 2019.
10. Gonzalez, R. C., Woods, R. E. Цифрова обробка зображень. — Pearson, 2018.
11. Lamba, A., Taneja, S. Практична веб-розробка за допомогою Symfony 5. — Apress, 2021.
12. Patterson, D. A., Hennessy, J. L. Організація та проєктування комп'ютера: інтерфейс апаратного/програмного забезпечення. — Morgan Kaufmann, 2020.
13. Grewal, B. S., Brar, M. S. Біотехнологія тварин: методи та застосування. — CRC Press, 2020.
14. Peters, D. Біометричні системи: технології, проєктування та оцінка продуктивності. — Springer, 2019.

15. Silberschatz, A., Korth, H. F., Sudarshan, S. Концепції системи баз даних. — McGraw-Hill Education, 2020.
16. Гудфеллоу, І., Бенджіо, Ю., Курвіль, А. Глибоке навчання. — MIT Press, 2016.
17. Doctrine ORM – [Електронний ресурс] – Режим доступу: <https://www.doctrine-project.org/projects/orm.html>
18. Документація Symfony – [Електронний ресурс] – Режим доступу: <https://symfony.com/doc/current/index.html>
19. Документація MySQL – [Електронний ресурс] – Режим доступу: <https://dev.mysql.com/doc/>
20. Посібник з PHP – [Електронний ресурс] – Режим доступу: <https://www.php.net/manual/en/>
21. Документація Git – [Електронний ресурс] – Режим доступу: <https://git-scm.com/doc>
22. Посібник з HTML W3Schools – [Електронний ресурс] – Режим доступу: <https://www.w3schools.com/html/>
23. Посібник з CSS W3Schools – [Електронний ресурс] – Режим доступу: <https://www.w3schools.com/css/>
24. Посібник з JavaScript W3Schools – [Електронний ресурс] – Режим доступу: <https://www.w3schools.com/js/>
25. Інструмент для створення діаграм Draw.io – [Електронний ресурс] – Режим доступу: <https://app.diagrams.net/>
26. ПРОЄКТУВАННЯ ER-ДІАГРАМИ – [Електронний ресурс] – Режим доступу: <https://nationalteam.worldskills.ua/skills/proektirovanie-er-diagrammy/>
27. Діаграма варіантів використання (UseCase diagram) – [Електронний ресурс] – Режим доступу: https://flexberry.github.io/ua/fd_use-case-diagram.html
28. ПРОЄКТУВАННЯ USE CASE ДІАГРАМИ. ВИЗНАЧЕННЯ ФУНКЦІОНАЛЬНИХ МОЖЛИВОСТЕЙ СИСТЕМИ – [Електронний

- ресурс] – Режим доступу:
<https://nationalteam.worldskills.ua/skills/proektirovanie-use-case-diagrammy-opredelenie-funktsionalnykh-vozmozhnostey-sistemy/>
29. What is Sequence Diagram? – [Електронний ресурс] – Режим доступу:
<https://www.visual-paradigm.com/guide/uml-unified-modeling-language/what-is-sequence-diagram/>
30. UML - Activity Diagrams – Tutorialspoint – [Електронний ресурс] – Режим доступу: https://www.tutorialspoint.com/uml/uml_activity_diagram.htm
31. Побудова діаграми класів – [Електронний ресурс] – Режим доступу:
https://flexberry.github.io/ua/gpg_class-diagram.html
32. UML-діаграми класів – [Електронний ресурс] – Режим доступу:
<https://prog-cpp.ua/uml-classes/>
33. Entity Relationship Diagram - Data Modeling – [Електронний ресурс] – Режим доступу:
<https://www.visualparadigm.com/VPGallery/datamodeling/EntityRelationshipDiagram.html>
34. UML 2 Tutorial - Package Diagram - Sparx Systems – [Електронний ресурс] – Режим доступу:
<https://sparxsystems.com/resources/tutorials/uml2/package-diagram.html>
35. Документація Bootstrap (офіційний сайт) – [Електронний ресурс] – Режим доступу: www.getbootstrap.com/documentation.

ДОДАТОК А**Фрагменти програмного коду. Функція створення тварини та
додання дерматогліфа**

```
class CattleController extends AbstractController
{
    #[Route('/cattles', name: 'app_cattle')]
    public function index(EntityManagerInterface $entityManager): Response
    {
        $cattles = $entityManager->getRepository(Cattle::class)->findAll();

        return $this->render('cattles/index.html.twig', [
            'cattles' => $cattles
        ]);
    }

    #[Route('/cattles/create', name: 'cattle_create')]
    public function create(Request $request, EntityManagerInterface $em): Response
    {
        $cattle = new Cattle();

        $farms = $em->getRepository(Farm::class)->findAll();

        if ($request->isMethod('POST')) {
            $cattle->setTagNumber($request->request->get('tagNumber'));
            $cattle->setBreed($request->request->get('breed'));
            $cattle->setSex((int) $request->request->get('sex'));
            $birthDate = $request->request->get('birthDate');
            $cattle->setBirthDate(new \DateTime($birthDate));
            $cattle->setCreatedAt(new \DateTimeImmutable());
        }
    }
}
```

```

$farmId = $request->request->get('farm');

$farm = $em->getRepository(Farm::class)->find($farmId);

$cattle->setFarm($farm);


$iconPath = $request->request->get('iconPath');

$cattle->setIconPath($iconPath);


$em->persist($cattle);

$em->flush();


$files = $request->files->get('dermatoglyphs');

if ($files) {

    foreach ($files as $file) {

        if ($file && $file->isValid()) {

            $filename = uniqid("", true) . '.' . $file->guessExtension();

            $file->move($this->getParameter('dermatoglyph_dir'), $filename);


            $dermatoglyph = new Dermatoglyph();

            $dermatoglyph->setCattle($cattle);

            $dermatoglyph->setImagePath('/uploads/dermatoglyphs/' . $filename);

            $dermatoglyph->setComment($request->request->get('comment'));

            $dermatoglyph->setCreatedAt(new \DateTimeImmutable());

            $em->persist($dermatoglyph);

        }

    }

    $em->flush();

```

```

    }

    flash()->success('Худобу успішно додано');

    return $this->redirectToRoute('app_cattle');
}

$iconFiles = [
    'cow.png', 'goat.png', 'hen.png', 'horse.png', 'pig.png', 'sheep.png'
];

return $this->render('cattles/create.html.twig', [
    'farms' => $farms,
    'icons' => $iconFiles
]);
}

#[Route('/cattles/{id}/dermatoglyphs', name: 'cattle_dermatoglyphs')]
public function dermatoglyphs(int $id, EntityManagerInterface $em): Response
{
    $cattle = $em->getRepository(Cattle::class)->find($id);

    if (!$cattle) {
        throw $this->createNotFoundException('Худоба не знайдена');
    }

    $dermatoglyphs = $cattle->getDermatoglyphs();

```

```

return $this->render('cattles/dermatoglyphs.html.twig', [
    'cattle' => $cattle,
    'dermatoglyphs' => $dermatoglyphs,
]);
}
}

```

ДОДАТОК Б

Фрагменти програмного коду. Функціонал створення події

```

class EventController extends AbstractController
{
    #[Route('/events', name: 'app_event')]
    public function index(EntityManagerInterface $entityManager): Response
    {
        $events = $entityManager->getRepository(Event::class)->findBy([], ['createdAt' => 'DESC']);

        return $this->render('events/index.html.twig', [
            'events' => $events,
        ]);
    }

    #[Route('/events/create', name: 'event_create', methods: ['GET', 'POST'])]
    public function create(Request $request, EntityManagerInterface $em): Response
    {
        $categories = $em->getRepository(EventCategory::class)->findAll();
        $cattleList = $em->getRepository(Cattle::class)->findAll();
    }
}

```

```

if ($request->isMethod('POST')) {

    $categoryId = $request->request->get('category');

    $cattleId = $request->request->get('cattle');

    $eventDate = $request->request->get('eventDate');

    $description = $request->request->get('description');


    $category = $em->getRepository(EventCategory::class)->find($categoryId);

    $cattle = $em->getRepository(Cattle::class)->find($cattleId);


    $event = new Event();

    $event->setCategory($category);

    $event->setCattle($cattle);

    $event->setUser($this->getUser());

    $event->setDescription($description);

    $event->setEventDate(new \DateTime($eventDate));

    $event->setCreatedAt(new \DateTimeImmutable());


    $em->persist($event);

    $em->flush();


    $this->addFlash('success', 'Подію успішно створено');

    return $this->redirectToRoute('app_event');

}

return $this->render('events/create.html.twig', [

    'categories' => $categories,

    'cattleList' => $cattleList,

]);

```

```

}

#[Route('/events/{id}', name: 'event_show', methods: ['GET'])]

public function show(Event $event): Response
{
    return $this->render('events/show.html.twig', [
        'event' => $event,
    ]);
}
}

```

ДОДАТОК В

НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ БІОРЕСУРСІВ І ПРИРОДОКОРИСТУВАННЯ УКРАЇНИ

Факультет інформаційних технологій

Декларація про академічну доброчесність та використання ШІ

ДЕКЛАРАЦІЯ ЗДОБУВАЧА

Я, Міхлевський Михайло Сергійович, здобувач НУБіП України, усвідомлюючи відповідальність за порушення академічної доброчесності, цією заявою підтверджую, що:

1. Моя бакалаврська кваліфікаційна робота на тему « Програмне забезпечення системи обліку та ідентифікації великої рогатої худоби за дерматогліфами» є результатом моєї особистої праці.
2. Усі запозичення з текстів інших авторів мають відповідні посилання згідно з чинними стандартами.
3. Щодо використання інструментів ШІ зазначаю, що:
 - ☐ інструменти генеративного ШІ не використовувалися.
 - ☐ інструменти ШІ ChatGPT були використані для:
 - ☐ перекладу іншомовних джерел;
 - ☐ стилістичного редагування та перевірки граматики;
 - ☐ допомоги у написанні/відлагодженні програмного коду;

▪ [] інше: _____.

Я розумію, що надання недостовірної інформації може призвести до скасування результатів захисту та відрахування з числа здобувачів НУБіП України.

«___» _____ 2026 р.

(підпис)
(підпис)

Михайло МІХЛЕВСЬКИЙ