Чернівецький національний університет імені Юрія Федьковича Факультет математики та інформатики

Кафедра математичного моделювання

**Лабораторна робота №2**

з навчальної дисципліни: “Проектування програмних систем” Написання UseCases

Виконав: студент 3 курсу

301 групи

спеціальності “Комп’ютерні науки”

Малованюк С.М

Перевірила: доцент Піддубна Л.А.

Варіант №26

**Чернівці – 2025**

**Завдання (предметна область - тваринна ферма).**

Відповідно до обраної предметної області на основі проведеного аналізу вимог замовника (лабораторна робота №1) розробити три різні usecases (по одному в короткій, поверхневій та повній формах відповідно) для свого проекту. Повна форма опису має містити всі пункти наведені в таблиці 1. Головний успішний сценарій повинен мати не менше 10 кроків. Передбачити не менше 5 альтернативних сценаріїв.

Use cases:

**I.** **Планування завдань та відстеження виконання робіт персоналом**

1. Планування завдань та відстеження виконання робіт персоналом (коротка форма).

Користувач (власник або адміністратор) вмикає комп’ютер, авторизується в системі, переходить до розділу «Керування завданнями», переглядає існуючі завдання або створює нове. Після введення даних система перевіряє їх коректність, зберігає завдання та надсилає сповіщення відповідальному співробітнику. У день виконання надходить нагадування. Після завершення роботи співробітник оновлює статус завдання та додає звіт за потреби. Власник або адміністратор переглядає звітність за потреби. У разі технічних проблем система сповіщає користувача та пропонує можливі рішення.

2. Планування завдань та відстеження виконання робіт персоналом (поверхнева форма).

*Головний сценарій (успішний):*

Користувач (власник або адміністратор) заходить на сайт із настільного комп’ютера чи іншого девайсу, що підключений до локальної мережі, вводить коректні логін та пароль, успішно авторизується в системі управління фермою та переходить до розділу управління завданнями. Він переглядає, за потреби - додає нове завдання, вказуючи назву, опис, відповідального працівника та дедлайн. Система перевіряє коректність введених даних та зберігає завдання. Співробітник отримує сповіщення про нове завдання, виконує його в зазначений термін та оновлює статус. Адміністратор за потреби переглядає виконане завдання та виходить із системи.

*Альтернативні сценарії:*

1. У користувача не завантажується сайт із системою.
2. Користувач не має доступ для додавання нових завдань.
3. Користувач ввів некоректні дані, що не завалідувалися для нового завдання.
4. Дані про завдання не зберігаються – система відображає відповідне повідомлення.
5. Співробітник не отримав сповіщення про нове завдання.

3. Планування завдань та відстеження виконання робіт персоналом (повна форма).

1. **Scope:** система управління тваринною фермою
2. **Level:** user-goal
3. **Primary Actor:** власник, адміністратор системи
4. **Stakeholders and Interests:  
   Власник:** бажає швидко та ефективно організувати робочі процеси, контролювати виконання поставлених завдань та генерувати аналітичні звіти для оптимізації роботи ферми.  
   **Адміністратор:** зацікавлений у можливості швидкого редагування розкладу, контролювання оновлення інформації та відстежуванням виконання завдань співробітниками.  
   **Працівники ферми:** зацікавлені отримувати чіткі завдання та вказівки, підтвердження, що їхня робота враховується.  
   **Ветеринари:** зацікавлені у своєчасному та актуальному інформуванні про розклад медичних оглядів та лікування тварин.  
   **Технічний персонал:** зацікавлений у безперебійній та безпечній роботі системи.
5. **Preconditions:  
   1)** Користувач успішно підключився до локальної мережі з допомогою електронного носія, успішно виконав авторизацію в системі, залогінившись в акаунт із відповідним рівнем доступу з допомогою особистого логіну та паролю, що надає відповідні можливості для взаємодії із завданнями.  
   **2)** Система містить актуальну інформацію про співробітників та попередні завдання.
6. **Main Success Scenario:  
   1) Вибір відповідного розділу:** Після авторизації користувач переходить до розділу «Керування завданнями».  
   **2)** **Перегляд існуючих завдань:** Система відображає список поточних завдань із зазначенням статусів (планується, виконано, у процесі).  
   **3) Створення нового завдання:** Користувач обирає опцію «Створити нове завдання».  
   **4)** **Введення даних завдання:** Система запитує введення відповідної інформації для створення завдання (назва, опис, дата й час, категорія завдання, відповідальний за завдання співробітник). **5)** **Валідація даних:** Система перевіряє правильність введених даних. **6)** **Підтвердження створення:** Після успішної валідації користувач натискає кнопку «Підтвердити» і завдання створюється. **7)** **Сповіщення співробітнику:** Система зберігає завдання в базі даних та надсилає відповідне сповіщення на акаунт користувача. **8)** **Нагадування у день виконання:** У встановлений день система автоматично надсилає нагадування співробітнику про заплановане завдання. **9) Підтвердження виконання:** Після завершення робіт співробітник входить до системи, змінює статус завдання на «Виконано» та за потреби додає короткий звіт про виконані роботи. **10)** **Користувач запитує звітність:** Система аналізує та оновлює інформацію у базі даних, формує звітність, готову до перегляду власником.
7. **Extensions:**a) У будь-який час виникає технічний збій:

1) Система виводить відповідне повідомлення.

2) Користувач очікує на автоматичне вирішення проблеми.

3) Система відновлює роботу.  
 3а) Якщо системі не вдалося відновити роботу автоматично:  
 -технічний персонал відновлює роботу системи

1a) Перейти до відповідного розділу не вдалось:

1. Користувач перепідключається до локальної мережі та оновлює сторінку.
2. Користувач продовжує свою роботу.

2a) Дані не були відображені:  
 1) Користувач перепідключається до локальної мережі та оновлює сторінку.  
 2) Користувач продовжує свою роботу.

3-6a) Відбулась помилка валідації певного/певних полів:  
 1) Система повідомляє користувача про помилку.  
 2) Користувач коректно заповнює поля та повторює спробу.

3-6b) Відбулась помилка у обробці графіку працівника:  
 1) Система повідомляє користувача про конфлікт із графіком працівника на обраний ним період часу.  
 2) Користувач приймає відповідне рішення та повторює спробу

3-6c)Відбулась помилка створення завдання:  
 1) Система повідомляє користувача про наявність ідентичного завдання.  
 2) Користувач знову переглядає список завдань та їхні статуси, знаходить потрібне завдання, за потреби редагує.  
 2a) Користувач не знаходить відповідне завдання:  
 - перезапускає систему  
 - повідомляє про проблему технічний персонал  
 - продовжує виконувати свою роботу

7a) Працівник не отримав сповіщення (не підтвердив його отримання):  
 1) Через певний час система дублює сповіщення.  
 2) При повторній відсутності підтвердження, система знову надсилає сповіщення працівнику, а також адміністратору.

8а) Працівник не отримав нагадування в день виконання:  
 1) Працівник сповіщає про це адміністрацію

9а) Працівник не підтвердив виконання завдання:  
 1) Працівник робить це пізніше із відповідною поміткою в системі.  
 2) Працівник повідомляє про це адміністрацію.

9b) Працівник не зміг підтвердити виконання завдання через технічні проблеми:  
 1) Працівник перевіряє з’єднання та оновлює сторінку.  
 2) Працівник перевірив, що він авторизований та повторює спробу.  
 2a) Працівник виявив, що він не авторизований та авторизуватись не може:  
 -система відображає відповідне повідомлення;  
 -працівник повідомляє про це адміністрацію.  
10a) Під час формування звітності відбувся конфлікт версій:  
 1) Система виводить: "Знайдено конфліктні зміни. Виберіть актуальну версію".  
 2) Користувач узгоджує зміни.  
 3) Система формує актуальну звітність

1. **Special Requirements:  
   1)** Інтерфейс повинен підтримувати роботу на мобільних пристроях (iOS/Android).  
   **2)** Система має автоматично генерувати резервні копії даних за заданим патерном та зберігати їх на окремому сервері.  
   **3)** Інформація про завдання, сповіщення і звітність повинні оновлюватися в режимі реального часу, що дозволяє оперативно реагувати на зміни у робочих процесах ферми.  
   **4)** Рішення має бути гнучким і здатним ефективно обробляти збільшення обсягу даних та кількості користувачів із розширенням ферми.
2. **Technology and Data Variations List:**1) Front-end: Реалізація інтерфейсу за допомогою технології React із застосуванням адаптивного дизайну для забезпечення зручного використання на різних пристроях.  
   2) Back-end і СУБД: Використання PostgreSQL та Node.js.  
   3) Аутентифікація: Підтримка багатофакторної автентифікації, що посилює захист доступу до системи та дозволяє розмежувати ролі (адміністратор, власник та інші).
3. **Frequency of Occurrence:** 95%.
4. **Miscellaneous (Open Issues):**  
   1) Провести аналіз зовнішніх факторів, наприклад впливу сезонних та погодних умов на графік завдань, а також оптимізація планування з урахуванням специфіки фермерської діяльності.  
   2) Дослідити більше можливостей відновлення роботи системи після аварійних ситуацій.

**II. Контроль запасів кормів**

1. Контроль запасів кормів (коротка форма).

Користувач (власник або адміністратор) вмикає комп’ютер та переходить на сайт, де авторизується в системі, переходить до розділу «Ресурси». Система відображає приблизну кількість поточних запасів кормів та перевіряє чи забезпечено належний рівень запасів, після чого може сповістити користувача про необхідність поповнення запасів.

2. Контроль запасів кормів (поверхнева форма).

*Головний сценарій (успішний):*

Власник переходить на сайт із пристрою, що підключений до локальної мережі, вводить коректні логін та пароль, успішно авторизується в системі управління фермою та переходить до розділу управління ресурсами. Система автоматично відображає приблизну кількість поточних запасів кормів, порівнює їх із встановленим мінімальним лімітом, рівень запасів у нормі, сповіщення про необхідність поповнення не відобразилось. Власник за потреби оновлює дані, генерує звіти та виходить із системи.

*Альтернативні сценарії:*

1. Після порівняння системою рівня запасів кормів із мінімальним лімітом – було виведено сповіщення про необхідність поповнення, оскільки рівень запасів підходить до критичного значення.
2. Дані про запаси кормів не завантажуються – система відображає відповідне повідомлення.
3. Власник помічає, що сповіщення спрацювало некоректно, він виявляє, що мінімальний ліміт було введено неправильно – власник змінює відповідні налаштування, і система переналаштовує сповіщення.
4. Сповіщення не спрацювало, хоча запаси на критичному рівні.
5. Власник вирішив оновити дані, але виникла помилка під час валідації даних.
6. Власник вирішив оновити дані, але виникла помилка під час збереження даних.
7. Власник вирішив згенерувати звіти, але дані для звітів не відображаються.

3. Контроль запасів кормі (повна форма).

1. **Scope:** система управління тваринною фермою
2. **Level:** user-goal
3. **Primary Actor:** власник або адміністратор системи
4. **Stakeholders and Interests:  
   Власник:** бажає мати стабільний спосіб підтримки належного рівня запасів кормів для забезпечення оптимального харчування тварин та оптимізації роботи ферми.  
   **Адміністратор:** зацікавлений у можливості вести точний облік кормів, своєчасно виявляти критичне зниження запасів та автоматизувати процес поповнення.  
   **Працівники ферми:** зацікавлені отримувати інформацію про зміни в рівнях запасів, що дозволяє їм коригувати робочий процес (наприклад, приготування корму чи його розподіл).  
   **Технічний персонал:** зацікавлений у безперебійній та безпечній роботі системи.
5. **Preconditions:  
   1)** Користувач успішно підключився до локальної мережі з допомогою електронного носія, успішно виконав авторизацію в системі, залогінившись в акаунт із відповідним рівнем доступу з допомогою особистого логіну та паролю, що надає відповідні можливості для взаємодії із системою.
6. **Main Success Scenario:  
   1) Вибір відповідного розділу:** Після авторизації користувач переходить до розділу «Ресурси».  
   **2)** **Перегляд поточних запасів:** Система відображає список поточних запасів.  
   **3)Аналіз запасів:** Система автоматично порівнює поточні запаси із встановленими мінімальними лімітами.  
   **4)Результат аналізу запасів:** Рівень запасів менше зазначеної норми, система відображає сповіщення про необхідність поповнення.  
   **5)** **Оновлення даних:** За потреби користувач обирає опцію оновлення даних запасів.  
   **6)Вибір категорії оновлення даних:** Система запитує напрямок зміни (надходження або списання) та відповідні числові значення.  
   **7)Валідація:** Система проводить валідацію введених даних.  
   **8)Підтвердження даних:** Після успішної валідації користувач натискає кнопку «Підтвердити», і система зберігає оновлені дані в базі. **9)Підтвердження встановлених лімітів:** Система просить підтвердження встановлених лімітів або ж їх оновлення. **10)Користувач запитує звітність:** Система оновлює звітність із зазначенням поточного стану запасів та генерує історичний звіт для аналітики.  
   **11)Завершення роботи:** Користувач завершує сесію та виходить із системи.
7. **Extensions:**a) У будь-який час виникає технічний збій:

1)Система виводить відповідне повідомлення.  
2)Користувач очікує на автоматичне вирішення проблеми.  
3)Система відновлює роботу.  
3а) Якщо системі не вдалося відновити роботу автоматично:  
 -технічний персонал відновлює роботу системи

b) Основний розділ «Ресурси» не завантажився або не відображається належним чином:

1) Система виводить відповідне повідомлення.  
2) Користувач перепідключається до локальної мережі та оновлює сторінку.  
3) Якщо проблема залишається, користувач повідомляє про проблему технічний персонал.  
2a) Користувач продовжує свою роботу після успішного оновлення.

1a) Перейти до відповідного розділу не вдалось:  
 1)Користувач перепідключається до локальної мережі та оновлює сторінку.  
 2)Користувач продовжує свою роботу.

2a) Дані не були відображені:  
 1) Користувач перепідключається до локальної мережі та оновлює сторінку.  
 2) Користувач продовжує свою роботу.

3-4a) Відбулась помилка аналізу запасів:  
 1) Система повідомляє користувача про відповідну помилку.  
 2) Користувач перевіряє відповідні дані, за потреби вносить корективи.  
 3) Користувач повторює спробу.  
 3-4b) Користувач помітив помилкове відображення сповіщення про поповнення:  
 1) Користувач перевіряє відповідні дані, за потреби вносить корективи.  
 3-4c) Сповіщення про поповнення запасів не спрацювало (ліміти було встановлено коректно):  
 1) Користувач перевіряє з’єднання до мережі.  
 2) Користувач сповіщає про це технічний персонал.  
 6-8a)Відбулась помилка валідації під час оновлення даних запасів:  
 1) Система виводить відповідне повідомлення .  
 2) Користувач повертається до форми оновлення даних та коригує помилкові поля.  
 3) Після корекції користувач повторює спробу.  
 6-8b)Відбулась помилка збереження даних під час їх оновлення:  
 1) Система виводить відповідне повідомлення .  
 2) Користувач повертається до форми оновлення даних, перевіряє дані та повторює спробу.  
 2а) Помилка повторюється:  
 -користувач повідомляє про це технічний персонал.  
 3) Якщо дублювання записів виявлено (наприклад, вже існує запис з таким же ідентифікатором), система виводить відповідне повідомлення.  
 4)Користувач приймає відповідне рішення та продовжує роботу.

9a)Відбулась помилка підтвердження лімітів:  
 1) Система виводить відповідне повідомлення .  
 2) Користувач перевіряє дані та повторює спробу.  
 2а) Помилка повторюється:  
 -користувач повідомляє про це технічний персонал.  
 10a)Функція генерації звітів не відображає дані:  
 1)Користувач перепідключається до локальної мережі та оновлює сторінку.   
 2)Користувач продовжує свою роботу.

1. **Special Requirements:  
   1)** Інтерфейс повинен підтримувати роботу на мобільних пристроях (iOS/Android).  
   **2)** Система має автоматично генерувати резервні копії даних за заданим патерном та зберігати їх на окремому сервері.  
   **3)** Інформація про завдання, сповіщення і звітність повинні оновлюватися в режимі реального часу, що дозволяє оперативно реагувати на зміни у робочих процесах ферми.  
   **4)** Рішення має бути гнучким і здатним ефективно обробляти збільшення обсягу даних та кількості користувачів із розширенням ферми.
2. **Technology and Data Variations List:**1) Front-end: Реалізація інтерфейсу за допомогою технології React із застосуванням адаптивного дизайну для забезпечення зручного використання на різних пристроях.  
   2) Back-end і СУБД: Використання PostgreSQL та Node.js.  
   3) Аутентифікація: Підтримка багатофакторної автентифікації, що посилює захист доступу до системи та дозволяє розмежувати ролі (адміністратор, власник та інші).
3. Frequency of Occurrence: 95%.
4. Miscellaneous (Open Issues):  
   1) Провести аналіз зовнішніх факторів, наприклад впливу сезонних та погодних умов на витрати кормів.  
   2) Дослідити більше можливостей відновлення роботи системи після аварійних ситуацій.  
   3) Дослідити варіанти покращень для автоматичної відправки запитів на закупівлю кормів.

**III. Моніторинг стану тварин**

1. Моніторинг стану тварин (коротка форма).

Користувач (власник, адміністратор або ветеринар) вмикає пристрій, підключається до мережі, авторизується в системі та переходить до розділу «Облік тварин». Він використовує пошук або фільтри для знаходження конкретної тварини чи групи тварин. Система відображає детальну інформацію про тварину, включаючи її стан здоров'я, історію хвороб, вакцинації та заплановані огляди. За потреби користувач може додати новий запис про стан здоров'я, оновити існуючі дані або запланувати ветеринарний захід. Система перевіряє та зберігає зміни, оновлюючи інформацію про тварину.

2. Моніторинг стану тварин (поверхнева форма).

*Головний сценарій (успішний):*

Користувач (ветеринар) заходить на сайт системи управління фермою з пристрою, підключеного до локальної мережі. Він вводить коректні логін та пароль, успішно авторизується та переходить до розділу «Облік тварин». Використовуючи пошук за ідентифікаційним номером, він знаходить потрібну тварину. Система відображає її картку з повною інформацією (вік, порода, історія здоров'я, вакцинації). Ветеринар додає новий запис про проведений огляд та призначене лікування. Система валідує введені дані, зберігає їх та оновлює медичну історію тварини. Ветеринар переглядає оновлену картку та виходить із системи.

*Альтернативні сценарії:*

1. У користувача не завантажується розділ «Облік тварин».
2. Система не знаходить тварину за введеним ідентифікатором.
3. Користувач не має доступу для редагування медичних записів цієї тварини.
4. Під час додавання нового запису виникла помилка валідації даних (наприклад, некоректний формат дати або відсутні обов'язкові поля).
5. Системі не вдалося зберегти новий запис про стан здоров'я (наприклад, через проблеми з базою даних).
6. При спробі запланувати захід (наприклад, вакцинацію) виникає конфлікт з іншими запланованими подіями для цієї тварини або відповідального персоналу.

3. Моніторинг стану тварин (повна форма).

1. **Scope:** система управління тваринною фермою
2. **Level:** user-goal
3. **Primary Actor:** користувач (власник, адміністратор або ветеринар).
4. **Stakeholders and Interests:  
   Власник:** зацікавлений у загальному огляді стану здоров'я поголів'я, відстеженні продуктивності, аналізі витрат на лікування та своєчасному виявленні проблем для оптимізації роботи ферми.  
   **Адміністратор:** зацікавлений у точності та актуальності даних про тварин, управлінні доступом до медичної інформації та забезпеченні стабільної роботи відповідного модуля системи.  
   **Ветеринари:** потребують доступу до повної та точної медичної історії кожної тварини для діагностики, планування лікування, профілактичних заходів (вакцинації, огляди) та відстеження ефективності терапії.  
   **Працівники ферми:** потребують доступу до основної інформації про стан здоров'я тварин для належного догляду, годування та своєчасного інформування ветеринара про помічені зміни.  
   **Технічний персонал:** зацікавлений у безперебійній та безпечній роботі системи.
5. **Preconditions:**1)Користувач успішно підключився до локальної мережі за допомогою свого пристрою.  
   2) Користувач успішно авторизувався в системі під своїм логіном та паролем, які надають йому відповідний рівень доступу до модуля «Облік тварин» та медичних записів.  
   3) Система містить у базі даних дані про тварин із їхніми ідентифікаторами та іншою інформацією.
6. **Main Success Scenario:  
   1) Вибір відповідного розділу:** Після авторизації користувач (Ветеринар) обирає в навігаційному меню розділ «Облік тварин».  
   **2)** **Відображення інтерфейсу:** Система відображає інтерфейс для роботи з даними тварин (список тварин, інструменти пошуку та фільтрації).  
   **3)Пошук тварини у системі:** Користувач вводить у поле пошуку унікальний ідентифікаційний номер тварини, яка потребує оновлення медичної картки.  
   **4)Результат пошуку:** Система здійснює пошук у базі даних та відображає знайдену тварину.  
   **5)Перехід до картки тварини:** Користувач клікає на запис тварини для переходу до детальнішої інформації про тварпну.  
   **6)Відображення повної інформації:** Система завантажує та відображає повну інформацію про тварину.  
   **7)Додавання записів:** Користувач обирає опцію для додавання записів, наприклад, медичних записів після огляду. Система відображає форму для введення нового медичного запису (дата огляду, симптоми і т.д.).  
   **8)Введення та валідація даних:** Користувач вводить необхідні дані, система проводить валідацію цих даних на коректність.  
   **9)Збереження даних:** Після успішної валідації користувач натискає відповідну кнопку для збереження даних, після чого система оновлює відповідну інформацію.  
   **10)Користувач переглядає оновлену інформацію:** Користувач переглядає оновлену картку тварини, переконуючись, що дані збережено коректно.  
   **11)Завершення роботи:** Користувач завершує сесію та переходить до інших завдань чи виходить із системи.
7. **Extensions:**

a) У будь-який час виникає технічний збій:

1)Система виводить відповідне повідомлення.  
2)Користувач очікує на автоматичне вирішення проблеми.  
3)Система відновлює роботу.  
3а) Якщо системі не вдалося відновити роботу автоматично:  
 -технічний персонал відновлює роботу системи

b) Основний розділ «Ресурси» не завантажився або не відображається належним чином:

1) Система виводить відповідне повідомлення.  
2) Користувач перепідключається до локальної мережі та оновлює сторінку.  
3) Якщо проблема залишається, користувач повідомляє про проблему технічний персонал.  
2a) Користувач продовжує свою роботу після успішного оновлення.

1a) Розділ «Облік тварин» не завантажується або працює некоректно:  
 1)Користувач перепідключається до локальної мережі та оновлює сторінку.  
 2)Користувач продовжує свою роботу.

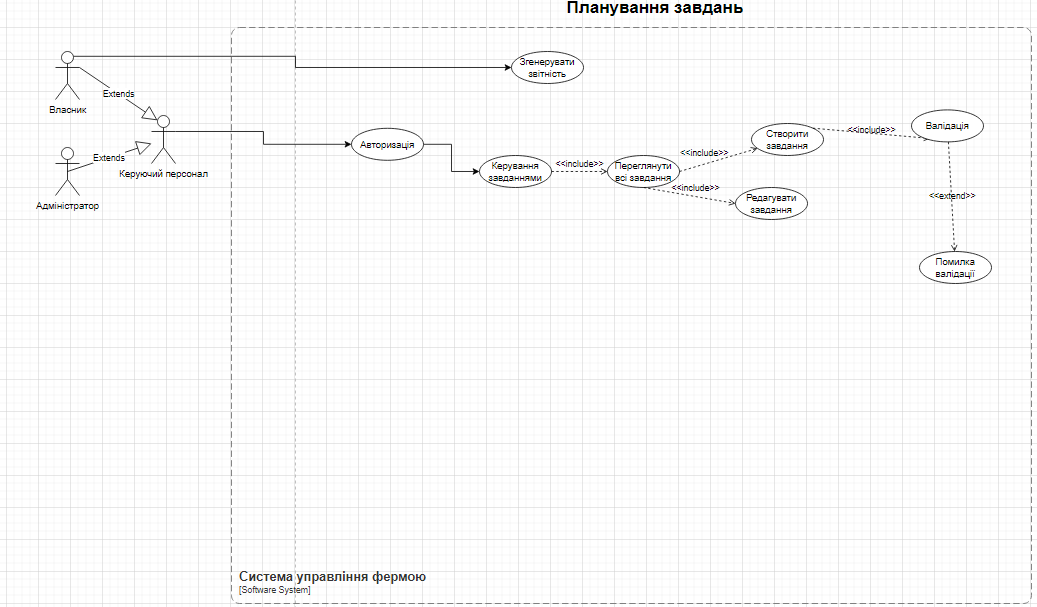
3a) Пошук за ідентифікатором не дав результатів:  
 1) Користувач перевіряє правильність введеного номера.  
 2) Користувач використовує інші критерії пошуку (наприклад, групу).  
 3) Якщо записи про тварину відсутні, користувач (за наявності прав) може додати їх та сповістити про це адміністрацію.  
6a) Дані в картці тварини завантажилися з помилкою або не повністю:  
 1) Система повідомляє користувача про відповідну помилку.  
 2) Користувач намагається оновити сторінку картки тварини.  
 3) Користувач повторює спробу.  
7a) Користувач не має прав на додавання/редагування медичних записів:  
 1) Кнопка/опція для додавання запису неактивна або система видає повідомлення про відсутність прав при спробі дії.  
 2) Користувач звертається до адміністратора для отримання необхідних прав, якщо це потрібно для його ролі.  
8a) Помилка валідації даних при заповненні медичної форми: 1) Система підсвічує поля з помилками та надає підказки щодо коректного формату або необхідних даних.  
 2) Користувач виправляє помилкові дані та повторює спробу.  
9a) Помилка збереження запису в базі даних:  
 1)Система відображає відповідне повідомлення.  
 2) Користувач перевіряє з'єднання та пробує зберегти ще раз через деякий час.  
 3)Якщо помилка є і надалі, то дані копіюються поза системою та повідомляє про проблему технічний персонал

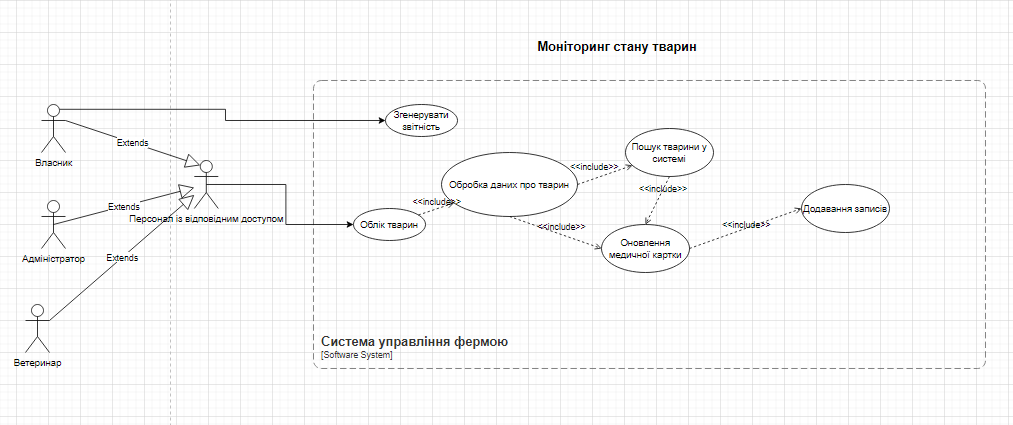
1. **Special Requirements:  
   1)** Інтерфейс повинен підтримувати роботу на мобільних пристроях (iOS/Android).  
   **2)** Система має автоматично генерувати резервні копії даних за заданим патерном та зберігати їх на окремому сервері.  
   **3)** Інформація про завдання, сповіщення і звітність повинні оновлюватися в режимі реального часу, що дозволяє оперативно реагувати на зміни у робочих процесах ферми.  
   **4)** Рішення має бути гнучким і здатним ефективно обробляти збільшення обсягу даних та кількості користувачів із розширенням ферми.
2. **Technology and Data Variations List:**1) Front-end: Реалізація інтерфейсу за допомогою технології React із застосуванням адаптивного дизайну для забезпечення зручного використання на різних пристроях.  
   2) Back-end і СУБД: Використання PostgreSQL та Node.js.  
   3) Аутентифікація: Підтримка багатофакторної автентифікації, що посилює захист доступу до системи та дозволяє розмежувати ролі (адміністратор, власник та інші).
3. **Frequency of Occurrence:** 95%.
4. **Miscellaneous (Open Issues):**  
   1) Провести аналіз зовнішніх факторів, наприклад впливу сезонних та погодних умов на здоров’я тварин.  
   2) Дослідити більше можливостей відновлення роботи системи після аварійних ситуацій.  
   3) Розглянути можливість інтеграції зі сканерами міток/чипів для швидкої ідентифікації тварин.

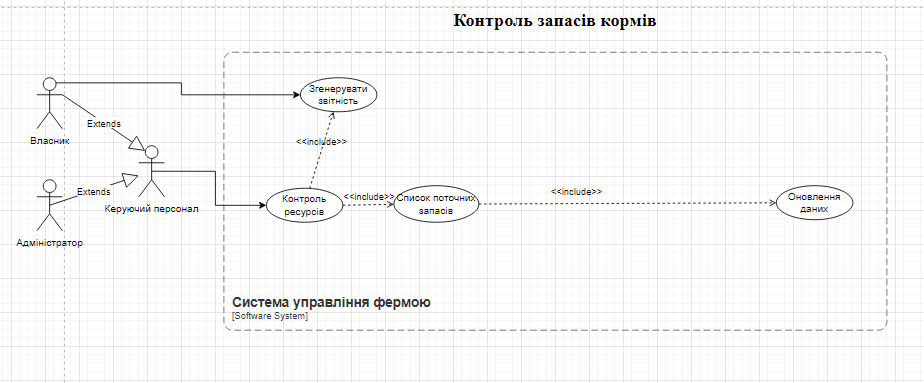
**Побудова діаграм UseCases**

**ЗАВДАННЯ**

У середовищі app.diagrams.net створити діаграму варіантів використання для обраного варіанта комп’ютерної системи. Діаграма повинна містити усіх акторів (користувачів системи) та по три варіанти використання для кожного актора. Пов’язати варіанти використання та акторів, при цьому використати усі види зв’язків (unidirectional association, generalization, extend relationship, include relationship).

1) 

2) 

3) 

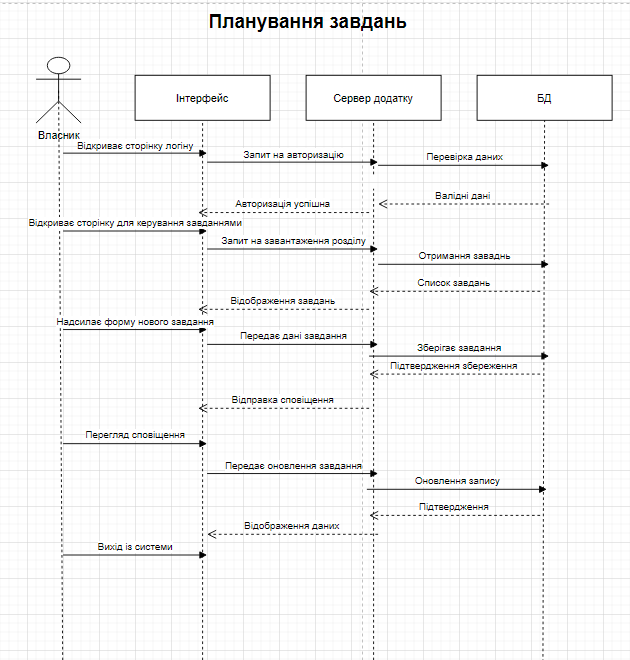
4) 

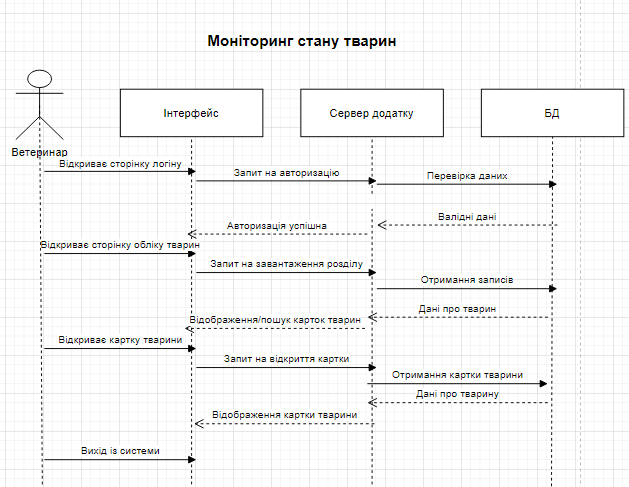
**Побудова діаграм взаємодії (Interaction Diagrams)**

**ЗАВДАННЯ**

Для кожного варіанта використання на Usecase Diagram створити Sequence або Collaboration Diagram (тобто у проекті повинно бути не менше шести діаграм кооперації та послідовності). На кожній діаграмі взаємодії повинен бути головний актор (при наявності) та не менше 5 об’єктів. Кожна діаграма взаємодії повинна містити не менше 10 повідомлень, якими обмінюються об’єкти в процесі виконання сценарію. Загальна сума різних об’єктів у проекті повинна налічувати 12-15 об’єктів. Об’єкти та повідомлення на діаграмах повинні мати зрозумілі назви. При побудові діаграм використовувати прямі, рефлексивні та зворотні типи повідомлень, а також символи знищення об’єктів.

1)



2)  


3)

