**University of Canberra**

**U3301315**

**Step 1: Understand and Define the Problem (Analyse)**

**Problem Statement**

The local animal shelter needs a **budget-friendly automated pet feeder** that can reliably serve both cats and dogs at specific times each day. Beyond just dispensing food, the system should also monitor whether the pets have eaten, detect any errors (such as food not being dispensed or left uneaten), and notify staff if something goes wrong. The overall design should remain simple and practical, using affordable components like servo motors and basic sensors, while being flexible enough for future upgrades.

**Key Features**

* Automatically dispense food at pre-set times (e.g., 8AM and 6PM).
* Track whether the food has been consumed using weight or infrared sensors.
* Alert staff if issues arise, such as an empty container or uneaten food.
* Allow schedules to be programmed or adjusted easily.

**Inputs**

* **Real-time clock** – to trigger feeding at scheduled times.
* **Food container level sensor** – to check if there is food available.
* **Bowl weight/infrared sensor** – to confirm if the food has been eaten.
* **Manual override button (optional)** – for staff to dispense food manually if needed.

**Outputs**

* **Servo motor control** – rotates to release food into the bowl.
* **Alert notifications** – simple buzzer, LED, or staff message if there’s an error.
* **Status display or log (optional)** – to track feeding activities.

**Assumptions & Limitations**

* The system only handles **dry pet food**.
* It serves **one pet at a time**.
* Built on a basic microcontroller with limited storage and memory.
* Cannot distinguish between cats and dogs (same schedule applies to both).
* Alerts are kept simple (e.g., buzzer or light) rather than advanced methods like SMS.

Type name purpose example value

Input clock triggers feeding time 8:00/ 16:00

Input food sensor check food is available empty / not empty

Input bowl sensor check pet ate food 0g, 20g,

Input manual button manual feed option pressed / not pressed

Output motor dispense food rotate / or not

Output alert warn staff problem on/ off

Output log/ display records feeding results success /fail

START

WHILE system is running:

Check current time

IF it is feeding time:

IF food is available:

Dispense food

Wait 10 minutes

IF pet has eaten:

Log "Success"

ELSE:

Alert "Pet did not eat"

ELSE:

Alert "Food empty"

ENDIF

END WHILE

STOP

| **Scenario** | **Input Condition** | **Expected Output** |
| --- | --- | --- |
| Pet eats as expected | Bowl weight decreases after 10 min | Log success, no alert |
| Pet does not eat | Bowl weight unchanged after 10 min | Trigger alert: "Pet did not eat" |
| Food bin is empty | Food level sensor = empty | Trigger alert: "Food container empty" |
| Manual feeding | Staff presses override button | Motor rotates, food dispensed |
| Feeding time mismatch | Current time ≠ schedule | No action |

**Possible Improvements**

* Send alerts to a mobile phone instead of just a buzzer.
* Allow different schedules for cats and dogs.
* Add a camera to monitor pets.
* Control portion size by adjusting motor time.

Part 3: AI Agent Integration – Reflection

**Prompt 1:** *“Review my Step 4 pseudocode and suggest improvements.”*  
**Response:** The AI made my pseudocode shorter and easier to understand. It replaced long sentences with simple actions like “Dispense food” and “Alert if empty.” This made my code clearer and more professional.

**Prompt 2: “Suggest possible improvements for the automated pet feeder system.”**

**Response:** Copilot came up with some great ideas to make the system more useful and realistic. It suggested adding mobile notifications so pet owners can get alerts when their pet is fed or when food is running low. It also recommended having separate feeding schedules for cats and dogs, which makes sense since their needs are different. Another smart idea was using a camera sensor to monitor the pet while feeding, and adding portion size control to avoid overfeeding. These suggestions helped me think beyond the basic design and imagine how the system could work better in everyday life.

**Prompt 3:** *“Help me write a professional README.md for my GitHub repository.”*  
**Response:** The AI provided a clear README structure with sections for project description, features, setup, and usage. This improved the organization and presentation of my GitHub repository.

Ethical Reflection

Copilot also helped me reflect on the ethical implications of automating pet care. It raised concerns about over-reliance on AI, potential malfunctions, and the importance of human oversight—points I included in my final report