A cover of a book

AI-generated content may be incorrect.

**Table of Contents**

[1. Understand and Define the Problem (Analyse) 3](#_Toc206349911)

[1.1. Understand the Problem. 3](#_Toc206349912)

[1.2. The features must the feeder include. 3](#_Toc206349913)

[1.3. Inputs 4](#_Toc206349914)

[1.4. Outputs 4](#_Toc206349915)

[1.5. Assumptions or Limitations 4](#_Toc206349916)

[1.6. Block Diagram (Sketch) 5](#_Toc206349917)

[2. Organise and Describe the Data. 6](#_Toc206349918)

[3. Plan the Solution (Design the Algorithm) 7](#_Toc206349919)

[4. Implement the Solution (Word Coding) 8](#_Toc206349920)

[5. Test and Refine the Solution (Debug and Verify) 9](#_Toc206349921)

[5.1. Improvements and Refinements 10](#_Toc206349922)

A person sitting at a computer

AI-generated content may be incorrect.

### Understand and Define the Problem (Analyze)

# Understand the Problem.

A person pulling a rope out of a head

AI-generated content may be incorrect.A local animal shelter needs a low-cost, programmable automated pet feeder for cats and dogs. The feeder must dispense food at scheduled times, check if food was dispensed and eaten (or how much was eaten), and alert staff if something goes wrong (eg: no food dispensed, food not eaten). The first output is the logic and behavior design that could later run on cheap hardware like a microcontroller with a servo motor and a few sensors.

# The features must the feeder include.

* Alert system

Notifies staff if there’s a problem (eg: no food dispensed, pet hasn’t eaten, food bin is empty).

* Simple event logging

Record feeding times, portion sizes, and alerts.

* Scheduled feeding times

Store up to 3 daily feeding times per feeder.

* Usage monitoring

Detect whether food was dispensed into the bowl and whether it was eaten

* Manual options to feed

Allow staff to feed pets manually in case of a system error.

# Inputs

* Feeding times (set by staff)
* Portion size
* Pet ID/type
* Real-time clock time/date
* Food storage level sensor
* Bowl weight sensor (grams before/after feed)
* Manual feed button

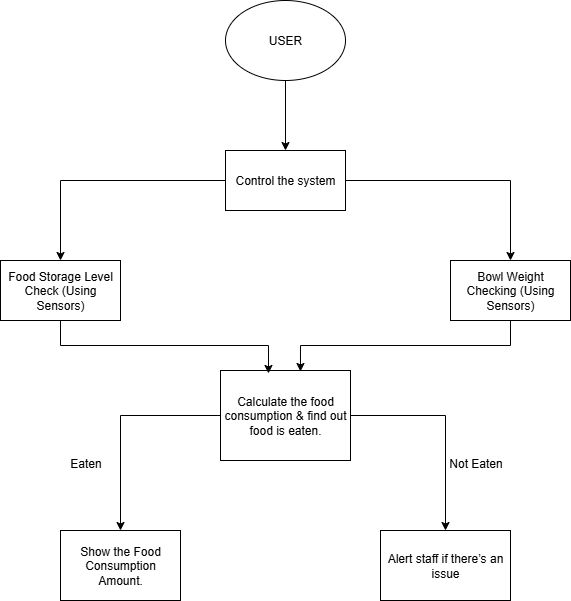
# Outputs

* Event log of feedings, consumption, and alerts
* Notify alerts, LED
* Servo motor control (to dispense food)

# Assumptions or Limitations

* Low-cost components
* Limited data storage
* One feeder serves one pet at a time (avoids multi-pet conflicts)
* Limited memory
* Max 3 feedings per day

# Block Diagram (Sketch)



### Organize and Describe the Data.

|  |  |  |  |
| --- | --- | --- | --- |
| **Data Type** | **Description** | **Sample Values** | **Operational Constraints** |
| Input: Real-world time & clock | The current time and date. | 2025.08.13 – 9.32 PM | It must be changed all the time to make sure it's correct. |
| Input: Food Level Sensor | An analog value indicating the amount of food in the main bin. | 75% | Must proceed below a certain level to notify staff. |
| Input: Bowl Weight Sensor | The current weight of the food bowl. | 100g (Empty Bowl)  245g (With Food) | Values will change based on what's in the bowl. |
| Input: Scheduled Times | The pre-programmed feeding times. | 7.30, 13.30 | Times are in a 24-hour format and can be updated by staff. |
| Output: Alert Signal | A command to trigger an alert system. | Turn on the Red LED, send a notification | If an error situation is met, like food not eaten or an empty bin, this alert will be triggered. |
| Parameter: Dispense Amount | The pre-set amount of food to be dispensed. | 50g | A set value that responds to the activity of the servo motor. |
| Parameter: Alert Time Limit | The amount of time to wait before alerting staff about uneaten food | 10 Minutes | If the bowl weight doesn't decrease within this time, an alert is sent to the staff. |

### A black background with white squares and dotsPlan the Solution (Design the Algorithm)

### Implement the Solution (Word Coding)

START

Check the current time

IF current time matches feeding schedule time THEN

IF the food bin is not empty THEN

Dispense food

Record initial bowl weight

WAIT 10 minutes

Record final bowl weight

IF final bowl weight < initial weight THEN

Display the amount of food that has been consumed

ELSE

Send an alert to the staff."

ENDIF

ELSE

Send an alert to refill the bin

ENDIF

ELSE

Wait until the next scheduled feeding

ENDIF

END

### Test and Refine the Solution (Debug and Verify)

|  |  |  |  |
| --- | --- | --- | --- |
| Scenario | Input | Output | Result |
| Pet Eats as Expected | Current\_time is 8:00, food\_bin\_level is full, initial\_bowl\_weight is 150g | The servo motor activates, and new\_bowl\_weight becomes 250g. After 10 minutes, final\_bowl\_weight is 150g. No alerts are sent. | The logic works correctly. It dispenses food, waits, and confirms the food has been eaten. |
| The pet does not eat | current\_time is 8:00, food\_bin\_level is full, initial\_bowl\_weight is 150g | The servo motor activates, and new\_bowl\_weight becomes 250g. After 10 minutes, final\_bowl\_weight is 250g. An alert is sent to staff. | The logic correctly identifies that the food hasn't been eaten and sends an alert to staff. |
| The food bin is empty | current\_time is 8:00, food\_bin\_level is empty | Display an error message to remind users to refill the food bin. | Logic handles the pre-dispensing check properly, stopping a failed attempt and telling staff to refill the bin. |

# Improvements and Refinements

* Add a camera for confirmation of eating
* Add a display to show system status and next feeding time
* Implement different portion sizes for cats and dogs
* Add battery backup for power outages
* Include WiFi connectivity for remote monitoring

A group of people holding a question mark and exclamation marks

AI-generated content may be incorrect.