UNIVERSITY OF CANBERRA  
INTRODUCTION TO INFORMATION TECHNOLOGY (4478/8936)

Assessment 1: The Solving Problem Process

Submitted to: Dr Julio Romero

Submitted by: Saima Shiraz

Student ID: 3317559

Step 1: Understand and Define the Problem (Analyse)

As of yet, there is no such system that both, dispenses food for the pets at scheduled times, and monitors whether the pets have eaten the food. The staff currently must do it manually for all the animals and is not aware if all of them have had their meal on time, and if they have, then how much the animals have had to eat. Therefore, the solution should be a low-cost, programmable automated pet feeder that should dispense the food at scheduled times and monitor whether the food has been consumed and the amount of food that has been consumed. This will also inform the staff of when the dispenser needs a refill, and of any alerts or issues regarding the system.

In this scenario, we are taking as assumption that only one pet will feed at a time, the feeding will occur only at the scheduled times, there will be a stable internet connection so that the feeder can run smoothly throughout the day, the feeder will also have internet connectivity to be able to send alerts regarding the refilling and errors to the staff, and that the portion sizes for each pet will already be set in advance.

Some examples of the inputs will be feeding times, real time clock, food sensors, weight sensors, and pet type, whereas some examples of the output will be alerts sent, log of feeding events, and the feeding status display.

Step 2: Organise and Describe the Data

|  |  |  |  |
| --- | --- | --- | --- |
| **Inputs** | **Operational Constraints** | **Outputs** | **Operational Constraints** |
| Feeding times | Max four times a day | Servo Motor Activation | Runs only if feeding conditions are met |
| Weight Sensors | Reads every five minutes after feeding | Alert notification | Sent in under fifteen minutes after the scheduled feeding |
| Food level Sensor | Minimum 15% required to start the machine | Display of feeding status | Updates every ten seconds |
| Pet Type | Must be set in system configuration | Log feeding events | Stored in daily log file |
| Real-time clock reading | Must sync daily |  |  |
| Manual feed command | Only if food level is low |  |  |

|  |  |  |  |
| --- | --- | --- | --- |
| **Variables** | **Type of Variable** | **Example** | **Operational Constraints** |
| Time | Discrete | 12pm | Constantly updated |
| Bowl weight before feeding | Continuous | 90 grams | Reads before feeding time |
| Bowl weight after feeding | Continuous | 0 grams | Reads every five minutes after feeding |
| Feeding times | Discrete | 9am, 12pm | Max 4 times a day |
| Type of pet | Discrete | Cat, Dog | Must be set in system configuration |
| Food level | Discrete | Full, low, empty | Minimum 15% required to start the machine |
| Alert status | Discrete | Empty, Not Eaten, OK | Cleared after staff comes to check |
| Manual staff override | Boolean | True, False | Requires staff confirmation |
| Has pet eaten | Discrete | Completed | Pending, completed, error |

Step 3: Plan the Solution (Design the Algorithm)

A diagram of a process

AI-generated content may be incorrect.

Step 4: Implement the Solution (Word Coding)

BEGIN

LOOP

Check current time

IF current time is feeding time, then

Proceed to feeding time

ELSE wait 1 minute and check again

END IF

Read food level sensors

IF food level >= 15%, then

Activate servo motor to dispense food

ELSE display manual feed command, then

send alert to staff (dispenser empty)

Go on standby

END IF

Check if food was dispensed

IF servo motor activated and food dispensed, then

Check bowl weight before feeding

Wait 15 minutes

Check bowl after feeding

ELSE try once more, then

display manual feed command

send alert to staff (food not dispensed)

Go on standby

END IF

Check if pet has eaten

IF bowl weight after feeding < bowl weight before feeding, then

Update feed log

Go on standby

ELSE alert staff (food not eaten)

Go on standby

END IF

END

Step 5: Test and Refine the Solution (Debug and Verify)

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Test Scenarios** | **Input Values** | **Expected Outcome** | **Actual Outcome** | **Result** | **Improvements/Notes** |
| Pet eats as expected | Time: 9am  Food level: >= 15%  Weight before feeding: 90 grams  Weight after:0 grams | Dispense food, update feed log | Dispense food, update feed log | Pass | Works as it should |
| Pet does not eat | Time: 9am  Food level: >= 15%  Weight before feeding: 90 grams  Weight after feeding: 90 grams | Alert staff (food not eaten) | Alert staff (food not eaten) | Pass | If staff doesn’t respond in 15 minutes, send another alert |
| Dispenser is empty | Time: 9am  Food level: <15% | Manual feed command, Alert staff (dispenser empty) | Manual feed command, Alert staff (dispenser empty) | Pass | Should add a low food in the dispenser alert at least half an hour before feeding time |
| Food not dispensed | Time: 9am  Food level: >= 15%  Servo activation failed | Manual feed command, Alert staff (food not dispensed) | Manual feed command, Alert staff (food not dispensed) | Pass | Add retry attempt before ending alert |
| Feeding time not yet reached | Time: 8:45am  Feeding time: 9am | Wait before checking again | Wait before checking again | Pass | None |

Reflection on AI Integration

After coming up with the data and inputs and outputs of the presented problem, I prompted copilot to check if my variables, inputs, and outputs were correct, to which it responded with arranging my data in a proper table as well as adding appropriate data on its own. I also prompted it to give me an example of a logic, so that I could have help making my own logic for this problem. Moreover, once I was done with my flowchart and logic, I prompted it to see if both the things matched and were mirrorring each other. It said that the two were aligned quite well, and provided suggestions for improvement, which included adding a retry loop before the alert sent to the staff regarding food not being dispensed. Through this I realised that adding a retry once loop would give the machine another chance to run and for any glitch to be resolved automatically before the staff must be alerted again. AI integration helped me make sense of the whole problem and how it can be solved step by step, especially since this is my first time solving a problem like this and making the logic and flowchart for it.

GitHub Repository Link

https://github.com/saimashiraz/pet-feeder-project.git