Automated Pet Feeder System

Project Overview

This project is a simulation of a low-cost, programmable \*\*automated pet feeder\*\* designed for a local animal shelter. The system is intended to:

- Dispense food for pets at scheduled times

- Monitor whether the food has been consumed

- Alert staff if food is not dispensed or not eaten

- Log feeding events for tracking and analysis

The solution was developed using the Solving Problem Process taught in the Introduction to Information Technology course at the University of Canberra.

Repository Structure

Step1\_Analysis/ # Problem definition, assumptions, inputs/outputs

Step2\_Data/ # Data tables and operational constraints

Step3\_Flowchart/ # Flowchart exported from Draw.io

Step4\_Word\_Code/ # Pseudocode implementation

Step5\_Testing/ # Test scenarios and results

AI Integration/ # Copilot prompts and reflection

README.md # Project summary and documentation

Problem Statement

There is currently no system that both dispenses food for pets at scheduled times and monitors consumption. Staff must manually feed animals and cannot track whether meals were eaten or how much was consumed. This project proposes a programmable feeder that automates feeding, monitors consumption, and alerts staff when issues arise.

Assumptions

- Only one pet feeds at a time

- Feeding occurs only at scheduled times

- Stable internet connection is available

- Portion sizes are pre-configured

- Alerts are sent via internet to staff

Inputs and Outputs

Inputs

- Feeding times

- Real-time clock

- Food level sensor

- Bowl weight sensor

- Pet type

Outputs

- Servo motor activation

- Feeding status display

- Alert notifications

- Feeding event log

Logic Summary

The system checks the current time against scheduled feeding times. If it’s time to feed and food level is sufficient, it activates the servo motor to dispense food. It then monitors the bowl weight before and after feeding to determine if the pet has eaten. Alerts are sent if food is not dispensed or not eaten.

Testing

Five test scenarios were run:

1. Pet eats as expected

2. Pet does not eat

3. Dispenser is empty

4. Food not dispensed

5. Feeding time not yet reached

All tests passed, with suggestions for retry logic and early alerts added.

AI Integration

Microsoft Copilot was used to:

- Validate inputs, outputs, and variables

- Refine pseudocode and flowchart logic

- Suggest retrying mechanisms and alert improvements

- Enhance documentation and structure

Reflection included in `/AI Integration`.

Submission Notes

- All files are organized by step

- Flowchart created using Draw.io and exported as PNG

- GitHub repository shared with tutor and lecturer

- Final ZIP file includes all required documents

Author

Saima Shiraz

Student ID: 3317559

University of Canberra

Course: Introduction to Information Technology (4478/8936)