**Assignment 1 – Automated pet feeder system**

***PART 1: On the Solving Problem Process***

**Step 1 – Understand and define the problem**

A local animal shelter needs a low-cost, programmable automated pet feeder system that dispenses food at scheduled times, monitors food consumption and alerts staff if something goes wrong.

**Features of the feeder**

1. The feeder must dispense food automatically.
2. It must detect whether the pet ate the food and notify staff if not eaten within a time limit.

**Inputs and outputs needed**

Inputs-: Feeding Time. (24-hr).

Portion Size. (grams).

Eating Duration. (minutes).

Outputs-: Alerts

**Assumptions / limitations**

Food-: One bowl per device.

Feeding Time.

Portion Size.

Eating Duration

Check eaten or not

Alert

Dispense and detect if eaten or not.

Input Process Decision Output

**Step 2 – Organize and describe the data**

|  |  |  |
| --- | --- | --- |
| **Input** | **Type** | **Description** |
| Feeding Time | Time | Scheduled times to dispense food. |
| Portion Size | Weight | Amount of food dispensed at a time. |
| Eating Duration | Time | Time given to eat the food. |
| Food Eaten Status | Boolean | Simulated result after eating duration. |
| Alert | Text | Alert sent to staff. |

**Step 3 – Design the algorithm**

**(**[**actual draw.io**](https://drive.google.com/file/d/1qd25dvrVxr_tOF8dOW7tVbgrf90HbE_W/view?usp=sharing)**)** (*please open with draw.io)*

A diagram of a food quality control

AI-generated content may be incorrect.

**Step 4 – Word coding**

Start

Input Feeding Time, Portion Size, Eating Duration

Wait until Feeding Time

Dispense Portion Size

Start Eating Duration Timer

Check Food Eaten Status

If no,

Send Alert to Staff, wait for next Feeding Time

If yes,

Output Record

End.

**Step 5 – Test and refine the solution**

|  |  |  |
| --- | --- | --- |
| **Test Case** | **Process** | **Output** |
| Pet eats as expected | Dispense Portion Size, Food Eaten | Output Record |
| Pet does not eat | Dispense Portion Size, Food Not Eaten | Send Alert |
| Food bin is empty | System can’t dispense food | Send Alert |

**Discussion of logic**

1. The logic works for pet eating and not eating scenarios.
2. It sends alerts when food is not eaten.
3. It sends alerts when bin is empty.

**Refinements**

1. Add low food sensor.
2. Improve alert system to SMS instead of local alerts.

***PART 2: On the Use of Technology***

Step 1 – Set up repository

Step 2 – Organize repository [(*GitHub Repository Link)*](https://github.com/dinara7753/Pet-Feeder-Project.git)

Step 3 - Document the work

***PART 3: On AI Agent Integration***

1. **Refine your logic or Word Code:**

In developing my automated pet feeder system, I asked Copilot to review my Step 4 implementation of word coding, which included inputs for feeding time, portion size and eating duration, followed by a sequence of dispensing, monitoring, and alerting. My prompt was:  
**“I have designed a solution, and I want you to review my Step 4 implementation of word coding for the system and suggest improvements or identify any potential issues.”**

Copilot responded with a detailed analysis, highlighting strengths such as logical flow and modular inputs, while also identifying areas for improvement. It suggested clarifying how food consumption is detected, adding a grace period before sending alerts, validating inputs, and logging both successful and missed feedings. These insights helped me refine my logic to make the system more robust and realistic for shelter conditions.

The revised flow includes sensor-based verification, input validation, and a retry mechanism enhancing both reliability and animal welfare. This feedback significantly influenced my final solution by encouraging a more thoughtful and error-resistant design. I’ve included screenshots of this interaction to demonstrate how Copilot’s suggestions directly shaped my implementation.

(182 words)

A screenshot of a computer program

AI-generated content may be incorrect.

1. **Generate alternative solutions:**

While refining my automated pet feeder system, I asked Copilot to suggest alternative approaches to improve the logic of my flowchart. My prompt was:  
**“Can you help me explore different ways to solve this problem or enhance the logic of my flowchart?”**

Copilot responded with several creative and practical suggestions. It proposed using a decision tree structure to simplify conditional logic, integrating sensor feedback loops to dynamically adjust feeding portions, and adding a fail-safe mechanism in case of hardware malfunction. One standout idea was to implement a modular design where each function timing, portion control, monitoring, and alerting could be independently tested and updated. This helped me visualize the system as a set of interdependent but flexible components.

Another enhancement was the use of color-coded flowchart branches to distinguish between normal operations and error-handling paths. This made the diagram more intuitive and easier to debug. Copilot also recommended simulating edge cases, such as missed feedings or sensor errors, to ensure the system behaves predictably.

These insights encouraged me to rethink my initial linear design and adopt a more adaptive, fault-tolerant structure. I’ve included screenshots of this exchange to show how Copilot’s suggestions directly influenced my improved flowchart.

(197 words)

A screenshot of a black and white screen

AI-generated content may be incorrect.

1. **Reflect on ethics and limitations:**

To prepare for presenting my automated pet feeder project, I asked Copilot to help me write a professional README.md file that clearly communicates the purpose, functionality, and structure of my solution. My prompt was:  
**“Can you help me write a professional README.md file that summarizes my project for presentation?”**

Copilot responded with a well-organized template that included sections such as Project Overview, Features, System Flow, Technologies Used, Setup Instructions and Future Improvements. It emphasized clarity and conciseness, helping me articulate the problem I was solving, ensuring consistent and monitored feeding for shelter animals and how my system addressed it through timed dispensing, sensor feedback, and alert mechanisms.

One key improvement was Copilot’s suggestion to include a visual flowchart and sample input/output scenarios to make the README more engaging and informative. It also helped me refine my language to sound more professional and confident, which was especially useful for the presentation context.

This guidance shaped both my written documentation and verbal explanation making my project easier to understand for evaluators and peers. I’ve included screenshots of this interaction to show how Copilot’s structured approach helped me elevate the quality of my final presentation materials.

(193 words)

A screenshot of a computer program

AI-generated content may be incorrect.