Macalalag, Marion Elizabeth P.
U3303144
Masters in Business Informatics
4478 Introduction to Information Technology Assignment 1

## Step 4: Implement the Solution (Word Coding)

Deliverable: Sequence of tasks with suitable explanations.

### Automated Pet Feeder Logic Flow

1. **Read Current Time**
   - Retrieve the system's current time.
2. **Check Feeding Schedule**
   - If the time is **8:00 AM** or **6:00 PM**, proceed to dispense food.
   - If not, keep the feeder inactive.
3. **Dispense Food**
   - Activate the server motor to dispense food into the bowl. Depending on the feeder - if dog feeder : Dog food, if cat feeder : cat food
4. **Verify Dispense Success**
   - Check if the food level in the container is **greater than 0%**.
     - If yes, store the current food level and bowl weight in the database.
     - If not, trigger an alert and send an error message to staff indicating the dispense failed.
5. **Check Pet Presence**
   - Detect if the pet is currently present.
     - If not present, trigger an alert to staff indicating the pet is missing.
     - If present, proceed to monitor consumption.
6. **Monitor Consumption**
   - If a pet is currently present, wait **10 minutes** after dispensing.
   - Retrieve the **starting bowl weight** from the database.
   - Measure the **current bowl weight**.
7. **Evaluate Consumption**
   - Calculate the difference between starting and current bowl weight.
   - If the weight is **unchanged (weight change = 0)**:
     - Trigger an alert and send an error message to staff indicating the food was not eaten.
   - If the weight has **decreased (weight change > 0)**:
     - Display a **green light** on the feeder.
     - Send a success message confirming food was consumed.

Macalalag, Marion Elizabeth P.
U3303144
Masters in Business Informatics
4478 Introduction to Information Technology Assignment 1

***Pseudocode***

```
# Step 1: Read current time
current_time = get_current_time()

# Step 2: Check if it's feeding time (8AM or 6PM)
if current_time in ["08:00", "18:00"]:

    # Step 3: Dispense food
    rotate_servo_to_dispense()

    # Step 4: Check food level after dispensing
    starting_food_level = read_food_container_level()
    starting_bowl_weight = read_bowl_weight()

    if food_level > 0:
        # Step 5: Store food level and bowl weight
        store_to_database(starting_food_level, starting_bowl_weight)
    else:
        # Step 6: Alert staff - dispense failed
        send_alert("Food dispense unsuccessful. Container is empty.")
        return

    # Step 7: Check pet presence
    if pet_present = not present :
        send_alert("Pet is missing during feeding time.")
     else
    # Step 8: Wait for pet to eat
    wait_minutes(10)
    # Step 9: Check if food was eaten
    initial_weight = starting_bowl_weight
    current_weight = read_bowl_weight()
    weight_difference = initial_weight - current_weight

    if weight_difference == 0:
        send_alert("Food was not touched by the pet.")
    else:
        display_green_light()
        send_success("Pet has eaten successfully.")

else:
    # Not feeding time, keep feeder inactive
    keep_feeder_empty()
END
```