Pet Care Management System

Mini Project

Name	**:** Mohammed Binshid A
Roll No	**:** 58
Course Name:	C programming
Date	**:** 12-07-2024

INTRODUCTION

Project overview:

The purpose of this project is developing a system for managing the care of pets and assistance to their owners. The system is designed to enable users to keep a history of pet's health, set reminders for regular vaccinations, as well as schedule grooming appointments.

Problem statement:

Handling pet care schedules by manual operation can be very involving and inaccurate. For a pet owner who wishes to keep track of their pets genealogical tree, diseases, or help them fight fleas effectively, there is a dependable platform.

Objective:

Create a machine-based system for managing pet cares so that you won't forget about their vaccines and other grooming needs.

System Requirements

Minimum Requirements for C Programming Code to Run:

➢Hardware Requirement:

- A computer with at least 4GB RAM
- 500MB of free disk space

➢Software Requirement:

- **Operating System:** Windows/Linux/MacOS
- **Compiler:** GCC or any C compiler
- **IDE**: Code: Blocks, Dev-C++, or any C IDE

Design and Development

❖ Program Logic:

The system’s functions were supposed to include processing modules for maintaining pet´s health archives, setting up reminders for necessary vaccinations, and booking care times. The modules allow users to feed new pet details and edit or delete existing ones. Therefore, if a vaccination or groom is approaching the system will warn them.

1. **Data Structures:**
   - **HealthRecord**: Stores the date and description of a health record.
   - **VaccinationReminder**: Stores the date and vaccine name for a vaccination reminder.
   - **GroomingAppointment**: Stores the date and service for a grooming appointment.
   - **Pet**: Stores the pet’s name, species, age, and arrays of health records, vaccination reminders, and grooming appointments.
2. **Global Variables:**
   - `pets`: An array to store up to 100 pets.
   - `petCount`: Keeps track of the number of pets added.

3. **Functions:**

   - **addPet()**: Adds a new pet to the `pets` array.
   - **addHealthRecord()**: Adds a health record to a specific pet.
   - **setVaccinationReminder()**: Sets a vaccination reminder for a specific pet.
   - **addGroomingAppointment()**: Adds a grooming appointment for a specific pet.
   - **displayPetInfo()**: Displays information about a specific pet.

❖ Pseudocode:

```
Start
Define constants:
   MAX_PETS = 100
   MAX_RECORDS = 100
   MAX_REMINDERS = 100
   MAX_APPOINTMENTS = 100
```

```
Define structures:
    HealthRecord:
        date: string[11]
        description: string[100]

    VaccinationReminder:
        date: string[11]
        vaccine: string[50]

    GroomingAppointment:
        date: string[11]
        service: string[50]

    Pet:
        name: string[50]
        species: string[50]
        age: integer
        healthRecords: array of HealthRecord[MAX_RECORDS]
        healthRecordCount: integer
        vaccinationReminders: array of VaccinationReminder[MAX_REMINDERS]
        vaccinationReminderCount: integer
        groomingAppointments: array of GroomingAppointment[MAX_APPOINTMENTS]
        groomingAppointmentCount: integer

Declare global variables:
    pets: array of Pet[MAX_PETS]
    petCount: integer = 0

Function addPet():
    If petCount >= MAX_PETS:
        Print "Maximum number of pets reached."
        Return
    End If
    Print "Enter pet's name: "
    Read pets[petCount].name
    Print "Enter pet's species: "
    Read pets[petCount].species
    Print "Enter pet's age: "
    Read pets[petCount].age
    Initialize pets[petCount].healthRecordCount to 0
    Initialize pets[petCount].vaccinationReminderCount to 0
    Initialize pets[petCount].groomingAppointmentCount to 0
    Increment petCount

Function addHealthRecord():
    Print "Enter pet's name: "
    Read petName
    For each pet in pets:
        If pet.name equals petName:
            If pet.healthRecordCount >= MAX_RECORDS:
                Print "Maximum number of health records reached."
                Return
            End If
            Print "Enter health record date (YYYY-MM-DD): "
            Read pet.healthRecords[pet.healthRecordCount].date
            Print "Enter health record description: "
            Read pet.healthRecords[pet.healthRecordCount].description
            Increment pet.healthRecordCount
            Return
        End If
    End For
    Print "Pet not found."

Function setVaccinationReminder():
    Print "Enter pet's name: "
    Read petName
    For each pet in pets:
        If pet.name equals petName:
            If pet.vaccinationReminderCount >= MAX_REMINDERS:
                Print "Maximum number of vaccination reminders reached."
                Return
            End If
            Print "Enter vaccination date (YYYY-MM-DD): "
            Read pet.vaccinationReminders[pet.vaccinationReminderCount].date
            Print "Enter vaccine name: "
            Read pet.vaccinationReminders[pet.vaccinationReminderCount].vaccine
            Increment pet.vaccinationReminderCount
            Return
        End If
    End For
    Print "Pet not found."

Function addGroomingAppointment():
    Print "Enter pet's name: "
    Read petName
    For each pet in pets:
        If pet.name equals petName:
            If pet.groomingAppointmentCount >= MAX_APPOINTMENTS:
                Print "Maximum number of grooming appointments reached."
                Return
            End If
            Print "Enter grooming appointment date (YYYY-MM-DD): "
            Read pet.groomingAppointments[pet.groomingAppointmentCount].date
            Print "Enter grooming service: "
            Read pet.groomingAppointments[pet.groomingAppointmentCount].service
            Increment pet.groomingAppointmentCount
            Return
        End If
    End For
    Print "Pet not found."

Function displayPetInfo():
    Print "Enter pet's name: "
    Read petName
    For each pet in pets:
```

```
 If pet.name equals petName:
   Print pet's name, species, and age
   Return
 End If
End For
Print "Pet not found."
End
```

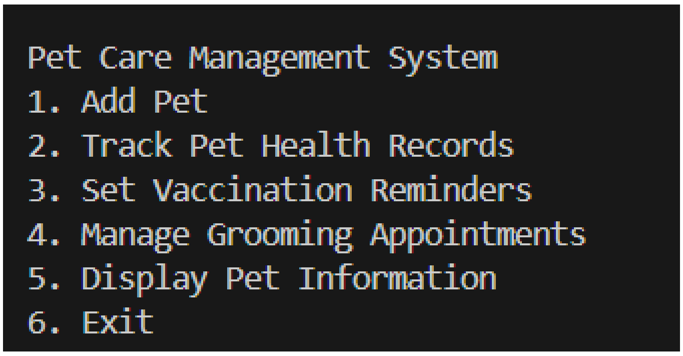
Testing and Results

❖ Test cases:

To add a new pet record and verify it; update an existing pet record and check the changes; delete a pet record and confirm its removal; set a reminder to ensure all changes were saved successfully.

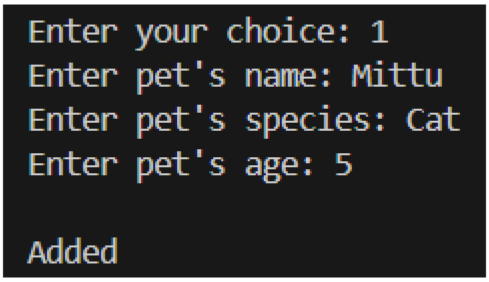
1. **Add a new pet record:**
    - Input: Mittu, Cat ,5
    - Expected Output: “Added”
2. **Update an existing pet health record:**
    - Input: Mittu, 2024-09-12, Good
    - Expected Output: “Health Condition updated”
3. **Set a vaccination reminder:**
    - Input: Mittu, 2024-12-01, Rabies
    - Expected Output: “Set vaccination reminder ”
4. **Add grooming appointment:**
    - Input: Mittu, 2024-12-05, Drying
    - Expected Output: “Set grooming appointment ”
5. **Display Pet Information:**
    - Input: Mittu
    - Expected Output: “Name: Mittu, Species: Cat, Age: 5
    - Health Records:
    - - 2024-09-12: Good
    - Vaccination Reminders:
    - - 2024-12-01: Rabies
    - Grooming Appointments:
    - - 2024-12-05: Drying”
6. **Exit:**
    - Input: Select “6”
    - Expected Output: “Exiting..”

Output:



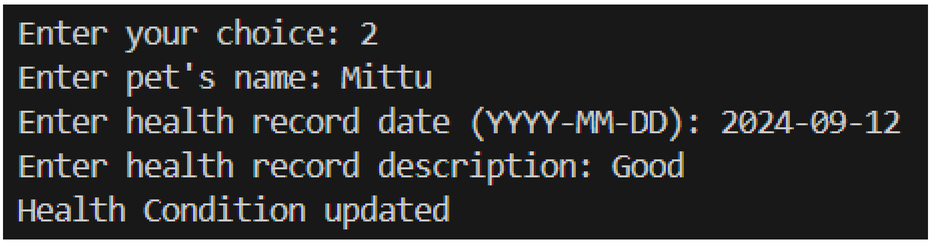
```
Pet Care Management System
1. Add Pet
2. Track Pet Health Records
3. Set Vaccination Reminders
4. Manage Grooming Appointments
5. Display Pet Information
6. Exit
```

Test case 1:



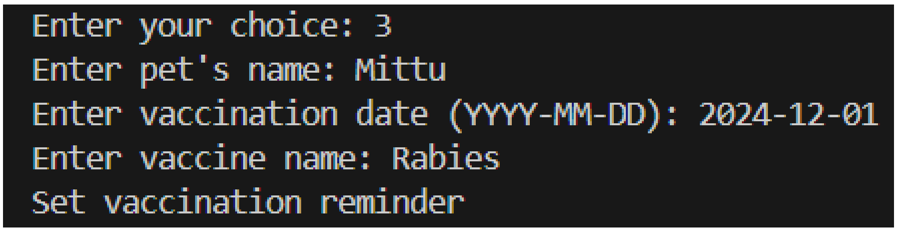
```
Enter your choice: 1
Enter pet's name: Mittu
Enter pet's species: Cat
Enter pet's age: 5

Added
```

Test case 2:



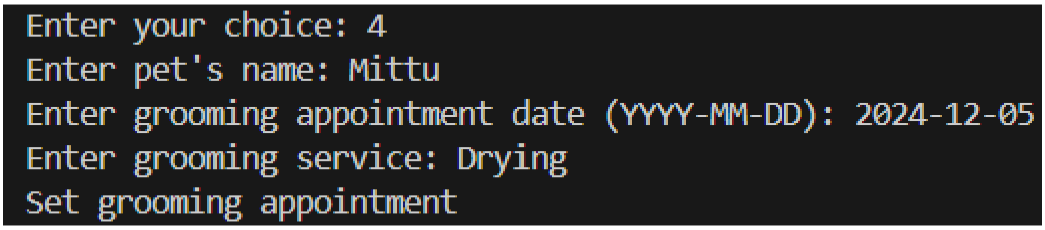
```
Enter your choice: 2
Enter pet's name: Mittu
Enter health record date (YYYY-MM-DD): 2024-09-12
Enter health record description: Good
Health Condition updated
```

Test case 3:



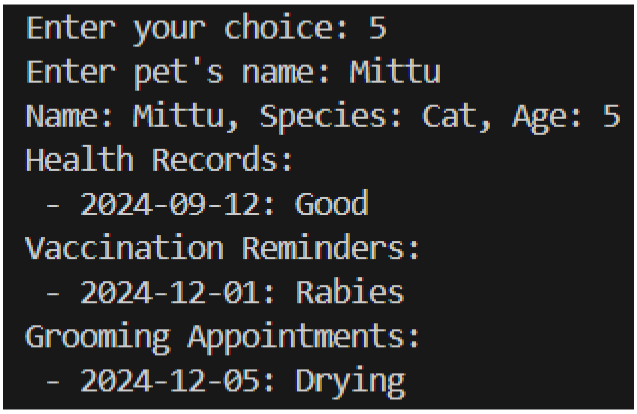
```
Enter your choice: 3
Enter pet's name: Mittu
Enter vaccination date (YYYY-MM-DD): 2024-12-01
Enter vaccine name: Rabies
Set vaccination reminder
```

Test case 4:



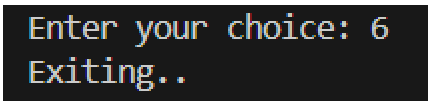
```
Enter your choice: 4
Enter pet's name: Mittu
Enter grooming appointment date (YYYY-MM-DD): 2024-12-05
Enter grooming service: Drying
Set grooming appointment
```

Test case 5:



```
Enter your choice: 5
Enter pet's name: Mittu
Name: Mittu, Species: Cat, Age: 5
Health Records:
 - 2024-09-12: Good
Vaccination Reminders:
 - 2024-12-01: Rabies
Grooming Appointments:
 - 2024-12-05: Drying
```

Test case 6:



```
Enter your choice: 6
Exiting..
```

Discussion of results

Managing statements and reminders for pets work well. Adding, updating or deleting files, as well as setting up necessary reminders, is very easy for the users.

Conclusion

➢Summary of the Project:

The Pet Care Management System offers an effective platform through which pet owners can monitor their pets' health records and care routines while tracking vaccinations and grooming appointments in a simplified manner.

➢Future Enhancements:

Features that might appear in forthcoming versions of the system include integration with veterinary clinics, mobile app support, and advanced notification systems.

References

https://www.sapphiresolutions.net/pet-management-app-development

https://capstoneguide.com/pet-care-management-system-capstone-project-document/

Appendices

Source code:

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#define MAX_PETS 100
#define MAX_RECORDS 100
#define MAX_REMINDERS 100
#define MAX_APPOINTMENTS 100
typedef struct {
    char date[11];
    char description[100];
} HealthRecord;
typedef struct {
    char date[11];
    char vaccine[50];
} VaccinationReminder;
typedef struct {
    char date[11];
    char service[50];
} GroomingAppointment;
typedef struct {
    char name[50];
    char species[50];
    int age;
    HealthRecord healthRecords[MAX_RECORDS];
    int healthRecordCount;
    VaccinationReminder vaccinationReminders[MAX_REMINDERS];
    int vaccinationReminderCount;
    GroomingAppointment groomingAppointments[MAX_APPOINTMENTS];
    int groomingAppointmentCount;
} Pet;
Pet pets[MAX_PETS];
int petCount = 0;
void addPet() {
    if (petCount >= MAX_PETS) {
        printf("Maximum number of pets reached.\n");
        return;
    }
    printf("Enter pet's name: ");
    scanf("%s", pets[petCount].name);
    printf("Enter pet's species: ");
    scanf("%s", pets[petCount].species);
    printf("Enter pet's age: ");
    scanf("%d", &pets[petCount].age);
    pets[petCount].healthRecordCount = 0;
    pets[petCount].vaccinationReminderCount = 0;
    pets[petCount].groomingAppointmentCount = 0;
    petCount++;
}
void addHealthRecord() {
    char petName[50];
    printf("Enter pet's name: ");
    scanf("%s", petName);
    for (int i = 0; i < petCount; i++) {
        if (strcmp(pets[i].name, petName) == 0) {
            if (pets[i].healthRecordCount >= MAX_RECORDS) {
                printf("Maximum number of health records reached.\n");
                return;
            }
            printf("Enter health record date (YYYY-MM-DD): ");
            scanf("%s", pets[i].healthRecords[pets[i].healthRecordCount].date);
            printf("Enter health record description: ");
            scanf(" %[^\n]", pets[i].healthRecords[pets[i].healthRecordCount].description);
            pets[i].healthRecordCount++;
            return;
        }
    }
    printf("Pet not found.\n");
}
void setVaccinationReminder() {
    char petName[50];
    printf("Enter pet's name: ");
    scanf("%s", petName);
    for (int i = 0; i < petCount; i++) {
        if (strcmp(pets[i].name, petName) == 0) {
            if (pets[i].vaccinationReminderCount >= MAX_REMINDERS) {
                printf("Maximum number of vaccination reminders reached.\n");
                return;
            }
            printf("Enter vaccination date (YYYY-MM-DD): ");
            scanf("%s",
pets[i].vaccinationReminders[pets[i].vaccinationReminderCount].date);
            printf("Enter vaccine name: ");
            scanf("%s",
pets[i].vaccinationReminders[pets[i].vaccinationReminderCount].vaccine);
            pets[i].vaccinationReminderCount++;
            return;
        }
    }
    printf("Pet not found.\n");
}
void addGroomingAppointment() {
    char petName[50];
    printf("Enter pet's name: ");
    scanf("%s", petName);
    for (int i = 0; i < petCount; i++) {
        if (strcmp(pets[i].name, petName) == 0) {
            if (pets[i].groomingAppointmentCount >= MAX_APPOINTMENTS) {
                printf("Maximum number of grooming appointments reached.\n");
                return;
            }
            printf("Enter grooming appointment date (YYYY-MM-DD): ");
            scanf("%s",
pets[i].groomingAppointments[pets[i].groomingAppointmentCount].date);
            printf("Enter grooming service: ");
            scanf("%s",
pets[i].groomingAppointments[pets[i].groomingAppointmentCount].service);
            pets[i].groomingAppointmentCount++;
            return;
        }
    }
    printf("Pet not found.\n");
}
void displayPetInfo() {
    char petName[50];
    printf("Enter pet's name: ");
    scanf("%s", petName);
    for (int i = 0; i < petCount; i++) {
        if (strcmp(pets[i].name, petName) == 0) {
            printf("Name: %s, Species: %s, Age: %d\n", pets[i].name, pets[i].species,
pets[i].age);
            printf("Health Records:\n");
            for (int j = 0; j < pets[i].healthRecordCount; j++) {
                printf(" - %s: %s\n", pets[i].healthRecords[j].date,
pets[i].healthRecords[j].description);
            }
            printf("Vaccination Reminders:\n");
            for (int j = 0; j < pets[i].vaccinationReminderCount; j++) {
                printf(" - %s: %s\n", pets[i].vaccinationReminders[j].date,
pets[i].vaccinationReminders[j].vaccine);
            }
            printf("Grooming Appointments:\n");
            for (int j = 0; j < pets[i].groomingAppointmentCount; j++) {
                printf(" - %s: %s\n", pets[i].groomingAppointments[j].date,
pets[i].groomingAppointments[j].service);
            }
            return;
        }
    }
    printf("Pet not found.\n");
}
void mainMenu() {
    int choice;
    while (1) {
        printf("\nPet Care Management System\n");
        printf("1. Add Pet\n");
        printf("2. Manage Pet Care Routines\n");
        printf("3. Track Pet Health Records\n");
        printf("4. Set Vaccination Reminders\n");
        printf("5. Manage Grooming Appointments\n");
        printf("6. Display Pet Information\n");
        printf("7. Exit\n");
        printf("Enter your choice: ");
        scanf("%d", &choice);
        switch (choice) {
            case 1:
                addPet();
                break;
            case 2:
                addHealthRecord();
                break;
            case 3:
                addHealthRecord();
                break;
            case 4:
                setVaccinationReminder();
                break;
            case 5:
                addGroomingAppointment();
                break;
            case 6:
                displayPetInfo();
                break;
            case 7:
                exit(0);
            default:
                printf("Invalid choice. Please try again.\n");
        }
    }
}

int main() {
    mainMenu();
    return 0;
}
```