sp24-pa6-Andy2Tran/src/main.cpp

```cpp
// Title  : main.cpp
// Desc   : Testing
// Name   : An Tran

#include "Address.h"
#include "Pet.h"
#include "Person.h"
#include <iostream>
    using std::cout;
    using std::cin;
    using std::getline;
#include <iomanip>
#include <vector>
#include <string>
#include <cstdlib>
#include <algorithm>
#include <limits>

void clearScreen() {
    #ifdef WINDOWS
        std::system("cls");
    #else
        std::system("clear"); // Assuming UNIX-based system
    #endif
}

int main() {
    std::vector<Person> people;
    int choice;

    do {
        clearScreen();
        cout << "Veterinary Clinic Management System\n"
                  << std::setfill('-') << std::setw(35) << std::setfill(' ') << '\n'
                  << "1. Enter a new person\n"
                  << "2. Enter a new pet for a person\n"
                  << "3. Delete a pet\n"
                  << "4. List all people and pets\n"
                  << "5. Exit\n";
        cout << "Enter your choice: ";
        cin >> choice;
        cin.ignore(); // Ignore the newline character left in the input buffer

        switch (choice) {
            case 1: {
                // Add a new person
                std::string fName, lName;
                std::string street, city, state; // to create address
                size_t zipCode;

                // Prompt for first name
                std::cout << "Enter first name: ";
                getline(cin, fName);
```

```
54
55                  // Prompt for last name
56                  std::cout << "Enter last name: ";
57                  getline(cin, lName);
58
59                  // Prompt for address details
60                  std::cout << "Enter street: ";
61                  getline(cin, street);
62
63                  std::cout << "Enter city: ";
64                  getline(cin, city);
65
66                  std::cout << "Enter state: ";
67                  getline(cin, state);
68
69                  std::cout << "Enter zip code: ";
70                  cin >> zipCode;
71                  cin.ignore(); // Ignore newline character
72
73                  Address address(street, city, state, zipCode);
74
75                  Person person(fName, lName, address);
76
77                  people.push_back(person);
78                  break;
79              }
80              case 2: {
81                  // Add a new pet to an existing person
82                  std::string fName, lName;
83                  cout << "Enter the first name of the person: ";
84                  getline(cin, fName);
85                  cout << "Enter the last name of the person: ";
86                  getline(cin, lName);
87
88                  auto it = std::find_if(people.begin(), people.end(), [&](Person&
    person) {
89                      return person.getFName() == fName && person.getLName() == lName;
90                  });
91
92                  if (it != people.end()) {
93                      cout << "Enter details for the new pet:\n";
94                      try {
95                          it->addPet();
96                      } catch (const std::runtime_error& e) {
97                          cout << e.what() << std::endl;
98                          cin.ignore(std::numeric_limits<std::streamsize>::max(), '\n')
    ;
99
100                     }
101                 } else {
102                     cout << "Person not found.\n";
103                 }
104                 break;
105             }
106             case 3: {
107                 // Delete a pet
108                 std::string fName, lName, petName;
```

```
109                 cout << "Enter the first name of the person: ";
110                 getline(cin, fName);
111                 cout << "Enter the last name of the person: ";
112                 getline(cin, lName);
113 
114                 auto it = std::find_if(people.begin(), people.end(), [&](Person&
    person) {
115                     return person.getFName() == fName && person.getLName() == lName;
116                 });
117 
118                 if (it != people.end()) {
119                     cout << "Enter the name of the pet to delete: ";
120                     getline(cin, petName);
121                     try {
122                         it->deletePet(petName);
123                     } catch (const std::runtime_error& e) {
124                         cout << e.what() << std::endl;
125                     }
126                 } else {
127                     cout << "Person not found.\n";
128                 }
129                 break;
130             }
131             case 4: {
132                 // List all people and their pets
133                 for (auto& person : people) {
134                     cout << person << "\n";
135                 }
136                 break;
137             }
138             case 5: {
139                 // Exit
140                 cout << "Exiting...\n";
141                 break;
142             }
143             default: {
144                 cout << "Invalid choice. Please try again.\n";
145                 break;
146             }
147         }
148 
149         cout << "Press Enter to continue...";
150         cin.get(); // Pause the program until the user presses enter
151 
152     } while (choice != 5);
153 
154     return 0;
155 }
156 
157 
158 
159 
```