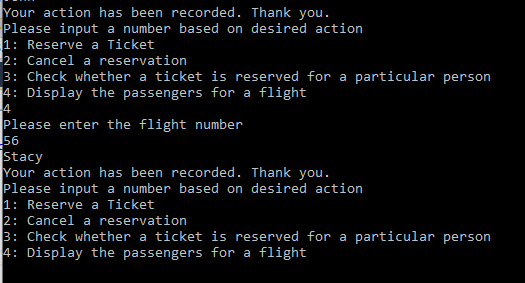
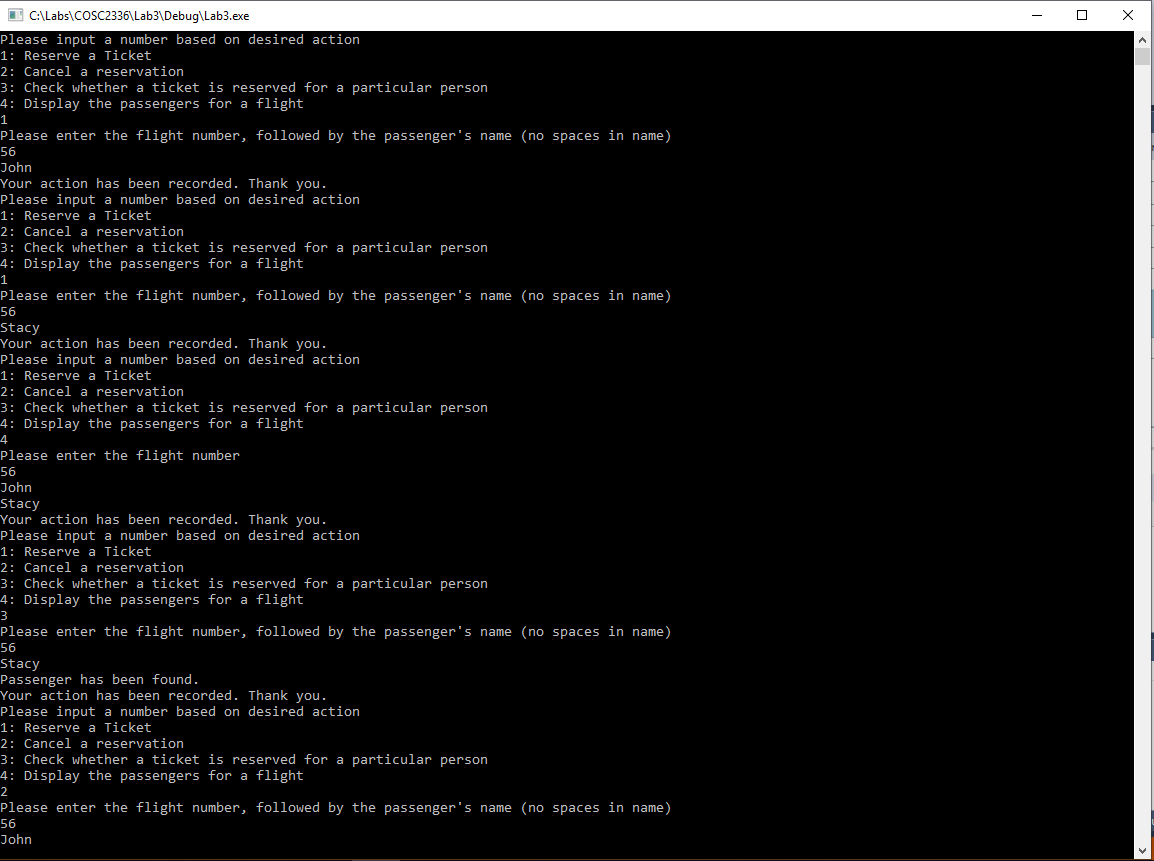
Name: William A. Brannon

Assignment: Lab Assignment Report #3

Date Due: Monday, February, 11, 2019

Class: Fundamentals of Programming III Section #1

**Program Output**



**Lab3.cpp**

// Lab3.cpp : linked lists

// By William A. Brannon on 01/23/2019

#include "stdafx.h"

#include <iostream>

#include <cctype>

#include <cstdlib>

#include <vector>

#include <list>

#include <algorithm>

#include <string>

using namespace std;

class Flight {

list<string> passengers;

public:

int id;

Flight() {

}

Flight(int num) {

id = num;

}

void addPassenger(string s) {

if (findPassenger(s)) {

}

else {

passengers.push\_back(s);

passengers.sort();

}

}

void removePassenger(string s) {

passengers.remove(s);

}

bool findPassenger(string s) {

for (string pass : passengers) {

if (pass == s) {

return true;

}

}

return false;

}

void displayPassengers() {

for (string pass : passengers) {

cout << pass << endl;

}

}

bool operator==(const Flight &other) const {

if (other.id == id && other.passengers == passengers) {

return true;

}

else {

return false;

}

}

};

class FlightController {

list<Flight> flights;

public:

FlightController() {

}

void addFlight(int i) {

bool found = false;

for (auto it = flights.begin(); it != flights.end(); it++) {

if (it->id == i) {

found = true;

}

}

if (found == false) {

Flight Temp = Flight(i);

flights.push\_back(Temp);

}

}

void removeFlight(int i) {

for (auto it = flights.begin(); it != flights.end(); it++) {

if (it->id == i) {

flights.erase(it);

break;

}

}

}

void addPassengerToFlight(int i, string s) {

addFlight(i);

for (auto it = flights.begin(); it != flights.end(); it++) {

if (it->id == i) {

int z = distance(flights.begin(), it);

it->addPassenger(s);

}

}

}

void removePassengerFromFlight(int i, string s) {

for (auto it = flights.begin(); it != flights.end(); it++) {

if (it->id == i) {

it->removePassenger(s);

}

}

}

bool findReservation(int i, string s) {

for (auto it = flights.begin(); it != flights.end(); it++) {

if (it->id == i) {

return it->findPassenger(s);

}

}

return false;

}

void displayPassengers(int i) {

for (auto it = flights.begin(); it != flights.end(); it++) {

if (it->id == i) {

it->displayPassengers();

}

}

}

void Menu() {

int x = 0;

cout << "Please input a number based on desired action" << endl;

cout << "1: Reserve a Ticket" << endl;

cout << "2: Cancel a reservation" << endl;

cout << "3: Check whether a ticket is reserved for a particular person" << endl;

cout << "4: Display the passengers for a flight" << endl;

cin >> x;

if (x == 1) {

int y;

string s;

cout << "Please enter the flight number, followed by the passenger's name (no spaces in name)" << endl;

cin >> y;

cin >> s;

addPassengerToFlight(y, s);

cout << "Your action has been recorded. Thank you." << endl;

}

else if (x == 2) {

int y;

string s;

cout << "Please enter the flight number, followed by the passenger's name (no spaces in name)" << endl;

cin >> y;

cin >> s;

removePassengerFromFlight(y, s);

cout << "Your action has been recorded. Thank you." << endl;

}

else if (x == 3) {

int y;

string s;

cout << "Please enter the flight number, followed by the passenger's name (no spaces in name)" << endl;

cin >> y;

cin >> s;

if (findReservation(y, s)) {

cout << "Passenger has been found." << endl;

}

else {

cout << "Passenger was not able to be found." << endl;

}

cout << "Your action has been recorded. Thank you." << endl;

}

else if (x == 4) {

int y;

cout << "Please enter the flight number" << endl;

cin >> y;

displayPassengers(y);

cout << "Your action has been recorded. Thank you." << endl;

}

else {

cout << "Invalid Input" << endl;

}

cin.clear();

cin.ignore(10000, '\n');

Menu();

}

};

int main()

{

FlightController controller;

controller.Menu();

system("pause");

return 0;

}