#include "stdafx.h"

#include <iostream>

#include <string>

#include <array>

#include <vector>

#include <conio.h>

#include <ctime>

using namespace std;

// START OF ERRORCHECK BEHAVIOURS

// Hides password details

string HidePasswordDetails()

{

char cInput = ' ';

string sPassword = "";

// Hides password details by outputting \* for each character inputted for the password

while (cInput != 13)

{

cInput = getch();

if (cInput != 13)

{

sPassword += cInput;

cout << "\*";

}

}

return sPassword;

}

// Checks if bad input has been entered such as letters or special characters, error checking

\_\_int16& OnlyIntegerErrorChecking(\_\_int16 &iNumberPass)

{

// Error checking if bad input is detected such as letters and special characters

while (cin.fail())

{

cin.clear(); // Clears bad input

cin.ignore(10000, '\n'); // Ignores characters until new line

cout << "Invalid input, please only enter numbers: \n";

cin >> iNumberPass; // Get new input

}

return iNumberPass;

}

// Checks if bad input has been entered such as letters or special characters, error checking for the phone number as \_\_int16 is too small for large number input

\_\_int32& PhoneNumberOnlyIntegerErrorChecking(\_\_int32 &iNumberPass)

{

// Error checking if bad input is detected such as letters and special characters

while (cin.fail())

{

cin.clear(); // Clears bad input

cin.ignore(10000, '\n'); // Ignores characters until new line

cout << "Invalid input, please only enter numbers: \n";

cin >> iNumberPass; // Get new input

}

return iNumberPass;

}

// Checks if an integer is 0 or less (negative numbers), error checking

\_\_int16 ZeroOrHigherErrorChecking(string sObjectTypePass, \_\_int16 iNumberPass)

{

// A minimum of 2 doctors must be entered into the program error checking

if (sObjectTypePass == "doctor")

{

// Error checking if the user has entered a number less than 2

while (iNumberPass < 2)

{

cout << "Invalid input, please only enter positive numbers, a minimum of 2: \n";

cin >> iNumberPass;

}

}

else

{

// Error checking if the user has entered a number less than 1

while (iNumberPass < 1)

{

cout << "Invalid input, please only enter positive numbers, a minimum of 1: \n";

cin >> iNumberPass;

}

}

return iNumberPass;

}

// Checks to make sure the user only selects from the options available, error checking

\_\_int16& MenuOptionErrorChecking(const string &sMenuTypePass, \_\_int16 &iChoicePass)

{

// Error checking for the first menu when signed in as a receptionist

if (sMenuTypePass == "SetupMenu")

{

while (iChoicePass < 1 || iChoicePass > 3)

{

cout << "Incorrect input detected, please select from the choices\n";

cin >> iChoicePass;

}

}

// Error checking for the second menu when signed in as a receptionist

else if (sMenuTypePass == "LoggedInMenu")

{

while (iChoicePass < 1 || iChoicePass > 14)

{

cout << "Incorrect input detected, please select from the choices\n";

cin >> iChoicePass;

}

}

// Error checking for the delete last object functions

else if (sMenuTypePass == "DeleteLastObject")

{

while (iChoicePass < 1 || iChoicePass > 2)

{

cout << "Incorrect input detected, please select from the choices\n";

cin >> iChoicePass;

}

}

return iChoicePass;

}

// Checks to make sure the user only selects from the options available for sugery and doctor, error checking

\_\_int16& OptionErrorChecking(\_\_int16 &iChoicePass, const \_\_int16 &iObjectSizePass)

{

// Loops until the choice number is greater that 1 and less than the object vector size

while (iChoicePass < 1 || iChoicePass > iObjectSizePass)

{

cout << "Invalid input, please only choose from the options provided: \n";

cin >> iChoicePass;

}

return iChoicePass;

}

// END OF ERRORCHECK BEHAVIOURS