Step 7 Screen Shots Project 2 Jairun Diemert

Screenshot 1 — Doctor.cpp

```cpp
//File: Doctor.cpp
//Author: Jairun Diemert
//Class: COP3014
//Project: 2
//Description: This is the member function file for Doctor.

#include <iostream>
#include <string>
using namespace std;

#include "Doctor.h"
#include "Appointment.h"

Doctor::Doctor(string name, int age, string specialty)
{
    nameD = name;
    ageD = age;
    specialtyD = specialty;
};
string Doctor::GetNameD()
{
    return nameD;
};
int Doctor::GetAgeD()
{
    return ageD;
};
string Doctor::GetSpecialtyD()
{
    return specialtyD;
};
Appointment * Doctor::GetOpenApp()
{
    return appointment;
};
```

Screenshot 2 — Patient.h

```cpp
//File: Patient.h
//Author: Jairun Diemert
//Class: COP3014
//Project: 2
//Description: This is the header file for the Patient class.

#ifndef PATIENT_H
#define PATIENT_H

#include <iostream>
#include <string>
using namespace std;

class Patient {
public:
    Patient(string name = "Empty", int age = 0, string sex = "Empty");
    string GetNameP();
    int GetAgeP();
    string GetSexP();
private:
    string nameP;
    int ageP;
    string sexP;
};
#endif
```

```cpp
//File: Patient.cpp
//Author: Jairun Diemert
//Class: COP3014
//Project: 2
//Description: This is the member function file for Patient.

#include <iostream>
#include <string>
using namespace std;

#include "Patient.h"

Patient::Patient(string name, int age, string sex)
{
    nameP = name;
    ageP = age;
    sexP = sex;
};
string Patient::GetNameP()
{
    return nameP;
};
int Patient::GetAgeP()
{
    return ageP;
};
string Patient::GetSexP()
{
    return sexP;
};
```

```
'Project2Diemert.exe' (Win32): Loaded 'C:\Windows\SysWOW64\sspicli.dll'. Cannot find or open the PDB file.
'Project2Diemert.exe' (Win32): Loaded 'C:\Windows\SysWOW64\cryptbase.dll'. Cannot find or open the PDB file.
'Project2Diemert.exe' (Win32): Loaded 'C:\Windows\SysWOW64\bcryptprimitives.dll'. Cannot find or open the PDB file.
'Project2Diemert.exe' (Win32): Loaded 'C:\Windows\SysWOW64\sechost.dll'. Cannot find or open the PDB file.
The thread 0x2740 has exited with code 0 (0x0).
The thread 0x19b0 has exited with code 0 (0x0).
The program '[6808] Project2Diemert.exe' has exited with code 0 (0x0).
```

```cpp
//File: Appointment.h
//Author: Jairun Diemert
//Class: COP3014
//Project: 2
//Description: This is the header file for the Appointment class.

#ifndef APPOINTMENT_H
#define APPOINTMENT_H

#include <iostream>
#include <string>
using namespace std;

class Appointment {
public:
    Appointment();
    Appointment(int timeScheduled, string patientName);
    int GetTimeScheduled();
    string GetPatientName();
    void SetPatientName(string patientName);
    void SetTimeScheduled(int timeScheduled);
    void AppointmentSchedular(string patientName, int timeScheduled, string doctorName);
    int DoctorLocator(string searchNameD);
    bool ScheduleOpen(int timeScheduled, int arrayLocationD);
    void ScheduleAppointment(int arrayLocationD, int timeScheduled, string patientName);
    void PrintPatients();
    void PrintDoctors();
    void PrintDoctorsANDAppointment();
    bool MenuPrompt();
    void RemoveAppointment(string doctor, int time);
    void CheckIfAppointmentOpen(string doctor, int time);
private:
    string patientNameP = "Empty";
    int timeScheduledP = 0;
};
#endif
```

```
'Project2Diemert.exe' (Win32): Loaded 'C:\Windows\SysWOW64\sspicli.dll'. Cannot find or open the PDB file.
'Project2Diemert.exe' (Win32): Loaded 'C:\Windows\SysWOW64\cryptbase.dll'. Cannot find or open the PDB file.
'Project2Diemert.exe' (Win32): Loaded 'C:\Windows\SysWOW64\bcryptprimitives.dll'. Cannot find or open the PDB file.
'Project2Diemert.exe' (Win32): Loaded 'C:\Windows\SysWOW64\sechost.dll'. Cannot find or open the PDB file.
The thread 0x2740 has exited with code 0 (0x0).
The thread 0x19b0 has exited with code 0 (0x0).
The program '[6808] Project2Diemert.exe' has exited with code 0 (0x0).
```

First screenshot code:

```cpp
//File: Appointment.cpp
//Author: Jairun Diemert
//Class: COP3014
//Project: 2
//Description: This is the member function file for Appointment.

#include <iostream>
#include<string>
using namespace std;

#include "PatientList.h"
#include "DoctorList.h"
#include "Appointment.h"

Appointment::Appointment()
{

};
Appointment::Appointment(int timeScheduled, string patientName)
{
    patientNameP = patientName;
    timeScheduledP = timeScheduled;
};
int Appointment::GetTimeScheduled()
{
    return timeScheduledP;
};
string Appointment::GetPatientName()
{
    return patientNameP;
}
void Appointment::SetPatientName(string patientName)
{
    patientNameP = patientName;
};
void Appointment::SetTimeScheduled(int timeScheduled)
{
    timeScheduledP = timeScheduled;
};
int Appointment::DoctorLocator(string searchNameD)
```

Second screenshot code:

```cpp
int Appointment::DoctorLocator(string searchNameD)
{
    for (int i = 0; i < DoctorList::size + 1; i++)
    {
        if (DoctorList::doctors[i].GetNameD() == searchNameD)
        {
            return i;
        }
    }
};
void Appointment::PrintPatients()
{
    for (int i = 0; i < PatientList::size; i++)
    {
        cout << "Name: " << PatientList::patients[i].GetNameP()
            << ", Age: " << PatientList::patients[i].GetAgeP()
            << ", Sex: " << PatientList::patients[i].GetSexP() << endl;
    }
};
void Appointment::PrintDoctors()
{
    for (int i = 0; i < DoctorList::size; i++)
    {
        cout << "Name: " << DoctorList::doctors[i].GetNameD()
            << ", Age: " << DoctorList::doctors[i].GetAgeD()
            << ", Specialty: " << DoctorList::doctors[i].GetSpecialtyD() << endl;
    }
};
void Appointment::PrintDoctorsANDAppointment()
{
    for (int i = 0; i < DoctorList::size; i++)
    {
        cout << endl;
        cout << "Name: " << DoctorList::doctors[i].GetNameD()
            << ", Age: " << DoctorList::doctors[i].GetAgeD()
            << ", Specialty: " << DoctorList::doctors[i].GetSpecialtyD() << endl;
        cout << "---------------------------------------" << endl;
        for (int j = 0; j < 8; j++)
        {
            int val = (DoctorList::doctors[i].GetOpenApp()[j].GetTimeScheduled());
```

```cpp
            int val = (DoctorList::doctors[i].GetOpenApp()[j].GetTimeScheduled());
            if ((val == 8) || (val == 9) || (val == 10) || (val == 11))
            {
                cout << " " << DoctorList::doctors[i].GetOpenApp()[j].GetTimeScheduled() << ":00AM    ";
                cout << DoctorList::doctors[i].GetOpenApp()[j].GetPatientName() << endl;
            }
            if ((val == 1) || (val == 2) || (val == 3) || (val == 4))
            {
                cout << " " << DoctorList::doctors[i].GetOpenApp()[j].GetTimeScheduled() << ":00PM    ";
                cout << DoctorList::doctors[i].GetOpenApp()[j].GetPatientName() << endl;
            }
            if (val == 0)
            {
                cout << " OPEN " << endl;
            }

        }
    }
};
bool Appointment::ScheduleOpen(int timeScheduled, int arrayLocationD)
{
    for (int i = 0; i < 8; i++)
    {
        if (DoctorList::doctors[arrayLocationD].GetOpenApp()[i].GetTimeScheduled() == timeScheduled)
        {
            return false;
        }
    }
    for (int j = 0; j < 8; j++)
    {
        if (DoctorList::doctors[arrayLocationD].GetOpenApp()[j].GetTimeScheduled() == 0)
        {
            return true;
        }
    }
    return false;
};
void Appointment::ScheduleAppointment(int arrayLocationD, int timeScheduled, string patientName)
{
    for (int i = 0; i < 8; i++)
    {
        if (DoctorList::doctors[arrayLocationD].GetOpenApp()[i].GetTimeScheduled() == 0)
        {
            DoctorList::doctors[arrayLocationD].GetOpenApp()[i].SetTimeScheduled(timeScheduled);
            DoctorList::doctors[arrayLocationD].GetOpenApp()[i].SetPatientName(patientName);
            cout << "Appointment booked." << endl;
            break;
        }
    }
};
void Appointment::AppointmentSchedular(string patientName, int timeScheduled, string doctorName)
{
    int arrayLocationD = DoctorLocator(doctorName);
    if (arrayLocationD < 6)
    {
        if (ScheduleOpen(timeScheduled, arrayLocationD))
        {
            ScheduleAppointment(arrayLocationD, timeScheduled, patientName);
        }
        else
        {
            cout << "Appointment not available at that time." << endl;
        }
    }
    else
    {
        cout << endl <<"Your entry does not match a doctor on the doctor list." <<endl;
        cout << "Please try again." << endl;
    }
};
void Appointment::RemoveAppointment(string doctor, int time)
{
    int val = -1;
    int arrayLocationD = DoctorLocator(doctor);
    if (arrayLocationD < 6)
    {
        for (int i = 0; i < 8; i++)
        {
            if (DoctorList::doctors[arrayLocationD].GetOpenApp()[i].GetTimeScheduled() == time)
```

Visual Studio — Project2Diemert

First window — Appointment.cpp

```cpp
157                     if (DoctorList::doctors[arrayLocationD].GetOpenApp()[i].GetTimeScheduled() == time)
158                     {
159                         DoctorList::doctors[arrayLocationD].GetOpenApp()[i].SetTimeScheduled(0);
160                         DoctorList::doctors[arrayLocationD].GetOpenApp()[i].SetPatientName("Empty");
161                         cout << "The appointment has been cleared" << endl;
162                         val = 2;
163                         break;
164                     }
165                 }
166                 if (val == -1)
167                 {
168                     cout << endl << "This appointment time is already available." << endl;
169                 }
170             }
171             if (arrayLocationD >= 6)
172             {
173                 cout << endl << "Your entry does not match a doctor on the doctor list." << endl;
174                 cout << "Please try again." << endl;
175             }
176         };
177     void Appointment::CheckIfAppointmentOpen(string doctor, int time)
178     {
179         int val = -1;
180         int arrayLocationD = DoctorLocator(doctor);
181         if (arrayLocationD < 6)
182         {
183             for (int i = 0; i < 8; i++)
184             {
185                 if (DoctorList::doctors[arrayLocationD].GetOpenApp()[i].GetTimeScheduled() == time)
186                 {
187                     cout << endl << "Appointment time not available." << endl;
188                     val = 2;
189                     break;
190                 }
191             }
192             if (val == -1)
193             {
194                 cout << endl << "Appointment time is OPEN." << endl;
195             }
196         }
197         if (arrayLocationD >= 6)
```

Second window — Appointment.cpp (MenuPrompt)

```cpp
193             {
194                 cout << endl << "Appointment time is OPEN." << endl;
195             }
196         }
197         if (arrayLocationD >= 6)
198         {
199             cout << endl << "Your entry does not match a doctor on the doctor list." << endl;
200             cout << "Please try again." << endl;
201         }
202     };
203     bool Appointment::MenuPrompt()
204     {
205         {
206             int time;
207             string patient;
208             string doctor;
209             string clear;
210             Appointment start;
211             bool end = false;
212             char option = '0';
213
214
215             cout << endl;
216             cout << "Please choose an option below" << endl;
217             cout << "-----------------------------" << endl;
218             cout << "1. Print doctor list." << endl;
219             cout << "2. Print patient list." << endl;
220             cout << "3. Schedule appointment." << endl;
221             cout << "4. Remove Appointment." << endl;
222             cout << "5. Print doctor list and the appointments scheduled." << endl;
223             cout << "6. Check if appointment time is available." << endl;
224             cout << "7. End Program." << endl;
225             cout << ": ";
226             cin >> option;
227             switch (option) {
228             case '1':
229                 cout << endl;
230                 start.PrintDoctors();
231                 cout << endl;
232                 system("pause");
```

Output:

```
'Project2Diemert.exe' (Win32): Loaded 'C:\Windows\SysWOW64\sspicli.dll'. Cannot find or open the PDB file.
'Project2Diemert.exe' (Win32): Loaded 'C:\Windows\SysWOW64\cryptbase.dll'. Cannot find or open the PDB file.
'Project2Diemert.exe' (Win32): Loaded 'C:\Windows\SysWOW64\bcryptprimitives.dll'. Cannot find or open the PDB file.
'Project2Diemert.exe' (Win32): Loaded 'C:\Windows\SysWOW64\sechost.dll'. Cannot find or open the PDB file.
The thread 0x2740 has exited with code 0 (0x0).
The thread 0x19b0 has exited with code 0 (0x0).
The program '[6808] Project2Diemert.exe' has exited with code 0 (0x0).
```

```
232              system("pause");
233              system("CLS");
234              break;
235          case '2':
236              cout << endl;
237              start.PrintPatients();
238              cout << endl;
239              system("pause");
240              system("CLS");
241              break;
242          case '3':
243              cout << endl <<"Please enter the patient name as it appears in the patient list." << endl;
244              cout << ":";
245              getline(cin, clear);
246              getline(cin, patient);
247              cout << endl << "Please enter the time you would like to book using the integer";
248              cout << " format from the list below." << endl;
249              cout << "~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~" << endl;
250              cout << "8  = 8:00AM" << endl;
251              cout << "9  = 9:00AM" << endl;
252              cout << "10 = 10:00AM" << endl;
253              cout << "11 = 11:00AM" << endl;
254              cout << "1  = 1:00PM" << endl;
255              cout << "2  = 2:00PM" << endl;
256              cout << "3  = 3:00PM" << endl;
257              cout << "4  = 4:00PM" << endl;
258              cout << ":";
259              cin >> time;
260              cout << endl <<"Please enter the doctor's name as it appears in the doctor list." << endl;
261              cout << ":";
262              getline(cin, clear);
263              getline(cin, doctor);
264              cout << endl;
265              start.AppointmentSchedular(patient, time, doctor);
266              cout << endl;
267              system("pause");
268              system("CLS");
269              break;
270          case '4':
271              cout << endl << "Please enter the appointment time you would like to remove using the integer";
```

```
271              cout << endl << "Please enter the appointment time you would like to remove using the integer";
272              cout << " format from the list below." << endl;
273              cout << "~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~" << endl;
274              cout << "8  = 8:00AM" << endl;
275              cout << "9  = 9:00AM" << endl;
276              cout << "10 = 10:00AM" << endl;
277              cout << "11 = 11:00AM" << endl;
278              cout << "1  = 1:00PM" << endl;
279              cout << "2  = 2:00PM" << endl;
280              cout << "3  = 3:00PM" << endl;
281              cout << "4  = 4:00PM" << endl;
282              cout << ":";
283              cin >> time;
284              cout << endl << "Please enter the doctor's name as it appears in the doctor list." << endl;
285              cout << ":";
286              getline(cin, clear);
287              getline(cin, doctor);
288              cout << endl;
289              start.RemoveAppointment(doctor, time);
290              cout << endl;
291              system("pause");
292              system("CLS");
293              break;
294          case '5':
295              cout << endl;
296              start.PrintDoctorsANDAppointment();
297              cout << endl;
298              system("pause");
299              system("CLS");
300              break;
301          case '6':
302              cout << endl << "Please enter the appointment time you would like check using the";
303              cout << " format from the list below." << endl << endl;
304              cout << "8  = 8:00AM" << endl;
305              cout << "9  = 9:00AM" << endl;
306              cout << "10 = 10:00AM" << endl;
307              cout << "11 = 11:00AM" << endl;
308              cout << "1  = 1:00PM" << endl;
309              cout << "2  = 2:00PM" << endl;
310              cout << "3  = 3:00PM" << endl;
```

First screenshot — Appointment.cpp:

```cpp
            cout << "11 = 11:00AM" << endl;
            cout << "1  = 1:00PM" << endl;
            cout << "2  = 2:00PM" << endl;
            cout << "3  = 3:00PM" << endl;
            cout << "4  = 4:00PM" << endl;
            cout << ":";
            cin >> time;
            cout << endl << "Please enter the doctor's name as it appears in the doctor list." << endl;
            cout << ":";
            getline(cin, clear);
            getline(cin, doctor);
            cout << endl;
            start.CheckIfAppointmentOpen(doctor, time);
            cout << endl;
            system("pause");
            system("CLS");
            break;
        case '7':
            end = true;
            break;
        default:
            cout << "Your entry was invalid please try again." << endl;
            break;
        }

        return end;
    }
};
```

Output (Debug):
```
'Project2Diemert.exe' (Win32): Loaded 'C:\Windows\SysWOW64\sspicli.dll'. Cannot find or open the PDB file.
'Project2Diemert.exe' (Win32): Loaded 'C:\Windows\SysWOW64\cryptbase.dll'. Cannot find or open the PDB file.
'Project2Diemert.exe' (Win32): Loaded 'C:\Windows\SysWOW64\bcryptprimitives.dll'. Cannot find or open the PDB file.
'Project2Diemert.exe' (Win32): Loaded 'C:\Windows\SysWOW64\sechost.dll'. Cannot find or open the PDB file.
The thread 0x2740 has exited with code 0 (0x0).
The thread 0x19b0 has exited with code 0 (0x0).
The program '[6808] Project2Diemert.exe' has exited with code 0 (0x0).
```

Second screenshot — DoctorList.h:

```cpp
/*
 *   File  : DoctorList.h
 *   Author: Jairun Diemert
 *   Course: COP3014
 *   Proj  : Project 02
 *   Descr : The declaration of the class that contains the initialized list
 *           of doctors available to be scheduled for the day.
 *
 */

#ifndef DOCTORLIST_H_
#define DOCTORLIST_H_

#include "Doctor.h"


class DoctorList
{
public:
    static Doctor doctors[];
    static const int size;
};


#endif /* DOCTORLIST_H_ */
```

```cpp
/*
 *    File  : DoctorList.cpp
 *    Author: Jairun Diemert
 *    Course: COP3014
 *    Proj  : Project 02
 *    Descr : The definition of the class that contains the initialized list
 *            of doctors available to be scheduled for the day.
 */


#include "Doctor.h"
#include "DoctorList.h"

Doctor DoctorList::doctors[] = { {"Gerardo Socia",       45, "Ophthalmology"},
                                 {"Jeremy Porche",       31, "Neurology"},
                                 {"Tamiko Poat",         57, "Pediatrics"},
                                 {"Marjorie Weatherman", 27, "Rheumatology"},
                                 {"Cheyenne Stevenson",  35, "Cardiology"} };

const int DoctorList::size = sizeof(doctors) / sizeof(Doctor);
```

```cpp
/*
 *    File  : PatientList.h
 *    Author: Jairun Diemert
 *    Course: COP3014
 *    Proj  : Project 02
 *    Descr : The declaration of the class that contains the initialized list
 *            of patients available to be scheduled for the day.
 */

#ifndef PATIENTLIST_H_
#define PATIENTLIST_H_

#include "Patient.h"

class PatientList
{
public:
    static Patient patients[];
    static const int size;
};

#endif /* PATIENTLIST_H_ */
```

```cpp
/*
 *    File : PatientList.cpp
 *    Author: Jairun Diemert
 *    Course: COP3014
 *    Proj : Project 02
 *    Descr : The definition of the class that contains the initialized list
 *            of patients available to be scheduled for the day.
 */

#include "Patient.h"
#include "PatientList.h"

Patient PatientList::patients[] = { {"Alan Harry",          24, "Male"},
                                    {"Ching Peppard",       32, "Male"},
                                    {"Lorelei Saad",        18, "Female"},
                                    {"Phoebe Schild",       23, "Female"},
                                    {"Tomasa Smither",      41, "Male"},
                                    {"Jessie Denicola",     40, "Female"},
                                    {"Keenan Hershberger",  69, "Male"},
                                    {"Apolonia Axtell",     52, "Other"},
                                    {"Bessie Boller",       78, "Female"},
                                    {"Logan Beverage",      64, "Male"},
                                    {"Toby Dill",           58, "Male"},
                                    {"Bryanna Luft",        67, "Female"},
                                    {"Fernande Weatherman", 74, "Male"},
                                    {"Marcell Coombes",     65, "Male"},
                                    {"Fidelia Vannest",     81, "Other"},
                                    {"Jaleesa Sylvest",     41, "Female"},
                                    {"Kelsie Watson",       61, "Female"},
                                    {"Analisa Manser",      39, "Female"},
                                    {"Pansy Haddock",       74, "Male"},
                                    {"Ervin Boltz",         26, "Male"},
                                    {"Brigida Huang",       38, "Female"},
                                    {"Tamie Magno",         46, "Female"},
                                    {"Marquerite Jozwiak",  16, "Female"},
                                    {"Mazie Hagwood",       70, "Other"},
                                    {"Alan Peppard",        24, "Male"},
                                    {"Ching Saad",          32, "Male"},
                                    {"Lorelei Schild",      18, "Female"},
                                    {"Phoebe Smither",      23, "Female"},
```

```cpp
                                    {"Pansy Haddock",       74, "Male"},
                                    {"Ervin Boltz",         26, "Male"},
                                    {"Brigida Huang",       38, "Female"},
                                    {"Tamie Magno",         46, "Female"},
                                    {"Marquerite Jozwiak",  16, "Female"},
                                    {"Mazie Hagwood",       70, "Other"},
                                    {"Alan Peppard",        24, "Male"},
                                    {"Ching Saad",          32, "Male"},
                                    {"Lorelei Schild",      18, "Female"},
                                    {"Phoebe Smither",      23, "Female"},
                                    {"Tomasa Denicola",     41, "Male"},
                                    {"Jessie Hershberger",  40, "Female"},
                                    {"Keenan Axtell",       69, "Male"},
                                    {"Apolonia Boller",     52, "Other"},
                                    {"Bessie Beverage",     78, "Female"},
                                    {"Logan Dill",          64, "Male"},
                                    {"Toby Luft",           58, "Male"},
                                    {"Bryanna Weatherman",  67, "Female"},
                                    {"Fernande Coombes",    74, "Male"},
                                    {"Marcell Vannest",     65, "Male"},
                                    {"Fidelia Sylvest",     81, "Other"},
                                    {"Jaleesa Watson",      41, "Female"},
                                    {"Kelsie Manser",       61, "Female"},
                                    {"Analisa Haddock",     39, "Female"},
                                    {"Pansy Boltz",         74, "Male"},
                                    {"Ervin Huang",         26, "Male"},
                                    {"Brigida Magno",       38, "Female"},
                                    {"Tamie Jozwiak",       46, "Female"},
                                    {"Marquerite Hagwood",  16, "Female"},
                                    {"Mazie Harry",         70, "Other"} };

const int PatientList::size = sizeof(patients) / sizeof(Patient);
```