HW 6 – Patient Management System

Contents

[1 Overview 1](#_Toc164001791)

[2 Requirements 2](#_Toc164001792)

[3 Submission 6](#_Toc164001793)

[4 Grading Criteria 6](#_Toc164001794)

# Overview

You will write a system that utilizes the *patients.mdb* database as shown on the right. You will use the Patients table first, then the Visits table, and then the Prescriptions table.

On the Schedule you will find the shell for the assignment that includes the complete GUI and the database. You will add data sources and code as required to fulfill the requirements in the next section.

**The shell also contains a spreadsheet where you will report the status and your estimate of the grade for each requirement.**

The shell was written in a previous version of VS. It should work on your computer (double-click the *.sln* file). If it doesn’t, there is only one page, *Default.aspx.* Thus, you can simply create a new project, add a *Default.aspx* web form, open it in Source mode, and the replace the code there, with the code in the shell (which you can open in notepad, or even VS). Note: I recommend replacing everything **except** the Page directive at the top.

I sampled 3 assignments from Spring 19, where each of the students earned an A and the times to complete were: 11, 11, and 11.5 hours. This is a pretty detailed assignment and relies completely on Labs 10, 11, and 12. It will take a lot of tinkering to get it to work.

I sampled 10 assignments from Spring 2020. The times ranged from 5 to 15 hours, with the average being 8.7 hours, for grades that ranged from 75 to 100.

# Requirements

1. All 3 GridViews for this assignment support sorting and pagination.
2. When you first visit the page, the patients are shown in a GridView as shown below.



1. A patient can be edited and deleted.
2. A patient can be added.

Hints: Use the approach from Lab 11. Remember to remove PatientID from the data source’s InsertCommand (don’t forget to remove a “?” also) and to remove the PatientID parameter. Alternately, you can use the manual approach from Lab 9.

1. When a patient is selected, the corresponding visits are shown in a GridView.

Hints: When configuring the DataSource: (a) do not select PatientID, (b) select Where clause and use PatientID from Patients GridView, (c) then choose Advanced and generate Insert, Update, Delete statements.



1. When a patient is selected, the total charges for all his/her visits is shown in the red label beside, “Total charges for selected patient”

Hints: Create a manual connection to the database (See Lab 9). Select all the charges for the selected patient. Get the PatientID from the gvPatients.SelectedValue. To read the charge from the data reader, use this to convert it to a double: (double)dr.GetDecimal(0).

An alternate approach is to use the SUM function in SQL, though you will have to learn this yourself.

* <https://www.w3schools.com/sql/sql_count_avg_sum.asp>
* <https://stackoverflow.com/questions/37122156/how-to-get-sum-of-numbers-using-sql-query-in-asp-net-c>
1. When a patient is selected, display the patient’s name in the red label beside, “Visits”.

Hints: (a) Create a SelectedIndexChanged event for Patients GridView, (b) grab names from SelectedRow (see Lab 11, or maybe 10?)

1. A visit can be edited and deleted.
2. A visit can be added for a selected patient.

Hints: Use the approach from Lab 11, however, it is a bit trickier. You need to remove the VisitID from the data source’s InsertCommand and to remove the VisitID parameter and replace it with PatientID in the InsertCommand and in the parameters. You can get the PatientID from gvPatients.SelectedValue. Alternately, you can use the manual approach from Lab 9.

1. When a visit is selected, the corresponding prescriptions are shown in a GridView.

Hints: When configuring DataSource: (a) do not select PatientID nor VisitID, (b) select Where clause and use VisitID from Visits GridView, (c) then choose Advance and generate Insert, Update, Delete statements



1. When a visit is selected, display the date of the visit in the red label beside “Prescriptions”.
2. A prescription can be edited and deleted.
3. A prescription can be added for a selected visit.

Hints: Use the approach from Lab 11, however, it is a bit trickier. Similar to adding a visit, but you need PatientID and VisitID.

1. When you try to delete a patient that has visits, an error message is displayed in a label.



1. When you try to delete a visit that has prescriptions, an error message is displayed in a label



# Submission

1. **Make sure you have completed the Status Report contained in the solution. Penalty=15 points**
2. Zip your **solution folder** into a file named: *hw06\_lastName.zip*.
3. Submit on Blazeview in the HW 6 dropbox.

# Grading Criteria

|  |  |
| --- | --- |
| **Overall Requirements** | **Worth** |
| 1 | All GridViews support sorting and pagination. | 5 |
| **Patient Actions** |  |
| 2 | When you first visit the page, the patients are shown in a GridView. | 10 |
| 3 | A patient can be edited and deleted. | 10 |
| 4 | When you try to delete a patient that has visits, an error message is displayed in a label. | 5 |
| 5 | A patient can be added. | 10 |
| 6 | When a patient is selected, the corresponding visits are shown in a GridView. | 10 |
| 7 | When a patient is selected the total charges for all his/her visits is shown in a label | 5 |
| 8 | When a patient is selected display their name in a label. | 5 |
| **Visit Actions** |  |
| 9 | A visit can be edited and deleted. | 5 |
| 10 | When you try to delete a visit that has prescriptions, an error message is displayed in a label | 5 |
| 11 | A visit can be added for a selected patient. | 5 |
| 12 | When a visit is selected, the corresponding prescriptions are shown in a GridView. | 10 |
| 13 | When a visit is selected, display the date of the visit in a label. | 5 |
| **Prescription Actions** |  |
| 14 | A prescription can be edited and deleted. | 5 |
| 15 | A prescription can be added for a selected visit. | 5 |
|  | **Total** | **100** |