

Cure Well - Hospital Management Web Application

Problem Statement:

"Cure Well" is a renowned hospital in Sydney. They require an application to plan surgeries based on the specialization and availability of doctors. You are expected to implement the Web Application using Angular, SQL Server, and Web API for the given requirements.

DATABASE

Database Design:

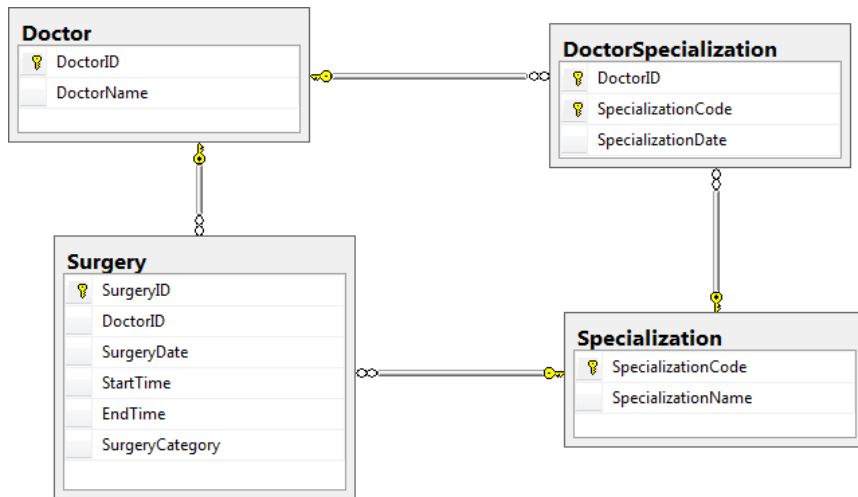


Fig: 01

a. Table Name: Doctor

This table contains the details of the doctors.

Field Name	Data Type	Description
<u>DoctorID</u>	INT	Primary key, IDENTITY, starts from 1001 and incremented by 1
DoctorName	VARCHAR(25)	Name of the doctor, Mandatory

Sample Data

DoctorID	DoctorName
1001	Albert
1002	Olivia
1003	Susan

b. Table Name: Specialization

This table contains the details of all specializations.

Fields:

Field Name	Data Type	Description
<u>SpecializationCode</u>	CHAR(3)	Primary key
SpecializationName	VARCHAR(20)	Name of the specialization, Mandatory

Sample Data:

SpecializationCode	SpecializationName
GYN	Gynecologist
CAR	Cardiologist
ANE	Anesthesiologist

c. Table Name: DoctorSpecialization

This table contains the specialization details of doctors.

Field Name	Data Type	Description
DoctorID	INT	Should be an existing doctor id in 'Doctor' table
SpecializationCode	CHAR(3)	Should be an existing specialization code in 'Specialization' table
SpecializationDate	DATE	Date on which specialization is added, Mandatory

Note: Combination of DoctorID and SpecializationCode is the primary key of the table

Sample Data:

DoctorID	SpecializationCode	SpecializationDate
1001	ANE	2010-01-01
1002	CAR	2010-01-01
1003	CAR	2010-01-01

d. Table Name: Surgery

This table contains the details of all the surgeries.

Field Name	Data Type	Description
SurgeryID	INT	Primary key, IDENTITY, starts with 5000 and auto-incremented by 1
DoctorID	INT	Should be an existing doctor id in 'Doctor' table
SurgeryDate	DATE	Date of the surgery, Mandatory
StartTime	DECIMAL(4,2)	Start time of the surgery, Mandatory
EndTime	DECIMAL(4,2)	End time of the surgery, Mandatory
SurgeryCategory	CHAR(3)	Should be an existing specialization code in 'Specialization' table

Sample Data:

SurgeryID	DoctorID	SurgeryDate	StartTime	EndTime	SurgeryCategory
5000	1001	2011-01-01	09.00	14.00	ANE
5001	1002	2015-01-01	10.00	16.00	CAR

Note: StartTime and EndTime should be in HH.MM (24 hours) format.

Data Access Layer Implementation Details

CureWellRepository
Class
Fields
context : CureWellContext
Properties
Context { get; } : CureWellContext
Methods
AddDoctor(Doctor dObj) : bool
CureWellRepository()
GetAllDoctors() : List<Doctor>
GetAllSpecializations() : List<Specialization>
GetAllSurgeryTypeForToday() : List<Surgery>
GetDoctorsBySpecialization(string specializationCode) : List<DoctorSpecialization>
UpdateDoctorDetails(Doctor dObj) : bool
UpdateSurgery(Surgery SObj) : bool

Action Name	Output/Return Value
GetAllDoctors	In case of success, returns list of doctors. In case of failure/exception, returns null.
GetAllSpecializations	In case of success, returns list of specializations. In case of failure/exception, returns null.
GetAllSurgeryTypeForToday	In case of success, returns list of surgery type for today In case of failure/exception, returns null.
AddDoctor	In case of success, returns true. In case of failure/exception, returns false.
UpdateDoctorDetails	In case of success, returns true. In case of failure/exception, returns false.
UpdateSurgery	In case of success, returns true. In case of failure/exception, false.
DeleteDoctor	In case of success, returns true. In case of failure/exception, false.

Note:

CureWell Data Access Layer (Dynamic Link Library) ASP.NET Web API Core project

The reference for Cure Well Data Access Layer (Dynamic Link Library) should be added to ASP.NET Web API

ASP.NET Web API Core

- The classes and properties for the Models are as follows

Doctor Class <div> Properties <ul style="list-style-type: none"> DoctorId { get; set; } : int DoctorName { get; set; } : string </div>	Specialization Class <div> Properties <ul style="list-style-type: none"> SpecializationCode { get; set; } : string SpecializationName { get; set; } : string </div>	Surgery Class <div> Properties <ul style="list-style-type: none"> DoctorId { get; set; } : int? EndTime { get; set; } : decimal StartTime { get; set; } : decimal SurgeryCategory { get; set; } : string SurgeryDate { get; set; } : DateTime SurgeryId { get; set; } : int </div>
DoctorSpecialization Class <div> Properties <ul style="list-style-type: none"> DoctorId { get; set; } : int SpecializationCode { get; set; } : string SpecializationDate { get; set; } : DateTime </div>		

Controller Implementation of Home

- The following actions should present in the 'HomeController'

Action Name	Description
GetDoctors	<ul style="list-style-type: none"> Gets all the doctor details by invoking the GetAllDoctors method of the data access layer and returns the same in Json format
GetSpecializations	<ul style="list-style-type: none"> Gets all the specialization details by invoking the GetAllSpecializations method of the data access layer and returns the same in Json format
GetAllSurgeryTypeForToday	<ul style="list-style-type: none"> Gets all the surgery details by invoking the GetAllSurgeryTypeForToday method of the data access layer and returns the same in Json format
AddDoctor	<ul style="list-style-type: none"> Adds the details of the doctor in the database by invoking the AddDoctor method of the data access layer Returns the status as true or false
UpdateDoctorDetails	<ul style="list-style-type: none"> Updates the details of the doctor in the database by invoking the UpdateDoctorDetails method of the data access layer Returns the status as true or false
UpdateSurgery	<ul style="list-style-type: none"> Updates the details of the surgery in the database by invoking the UpdateSurgery method of the data access layer Returns the status as true or false
DeleteDoctor	<ul style="list-style-type: none"> Deletes the doctor from the doctor details table by invoking DeleteDoctor method of the data access layer Returns the status as true or false

Angular

- Implement exception handling in all the methods

Implementation of CureWellService

- The following methods in 'CureWell.service.ts' need to be implemented.

Action Name	Description
getDoctors	• Should consume the appropriate web API action to get the details of all the doctors
getAllSpecializations	• Should consume the appropriate web API action to get all the specializations
getSurgeriesForToday	• Should consume the appropriate web API action to get all the surgeries scheduled for today
addDoctor	• Should consume the appropriate web API action to add a doctor
editDoctorDetails	• Should consume the appropriate web API action to edit the details of a doctor
editSurgery	• Should consume the appropriate web API action to edit the surgery time for a given surgery
deleteDoctor	• Should consume the appropriate web API action to delete the details of a doctor
errorHandler	• Should be called in case of any exception in the curewell.service.ts

Component Implementation of view-doctor

The following methods in 'view-doctor.component.ts' need to be implemented.

Method Name	Description
ngOnInit	• Invoke the getDoctor() method of the ViewDoctorComponent
getDoctor	<ul style="list-style-type: none">• Invoke the appropriate method of curewell.service.ts to fetch the details of all the doctors• In case of successful execution of service:<ul style="list-style-type: none">▪ Store the response in the variable 'doctorList'▪ Set the 'showMsgDiv' variable appropriately• In case of error<ul style="list-style-type: none">▪ Set the variable 'doctorList' to null▪ Store the response in the variable 'errorMsg'• On successful completion, display 'Doctors Fetched Successfully' on the console
editDoctorDetails	<ul style="list-style-type: none">• On click of 'Edit Doctor Details' button in view-doctor.component.html, this method must be invoked by passing the required parameters• Navigate to the UpdateDoctorComponent by passing the required parameters
removeDoctor	<ul style="list-style-type: none">• On click of 'Remove Doctor Details' button in view-doctor.component.html, this method must be invoked by passing the required parameters• Should delete the details of a doctor by invoking the appropriate method of curewell.service.ts• If the delete is successful,<ul style="list-style-type: none">▪ Store the response in the variable 'status'▪ If the response is true, display the message 'Doctor detailed deleted successfully!' using alert box.▪ Else display 'Doctor's name not deleted' using alert box.• In case of any error<ul style="list-style-type: none">▪ Set the variable 'errorMsg' to 'Some error occurred'

Component Implementation of add-doctor

- The following methods in 'add-doctor.component.ts' need to be implemented

Method Name	Description
addDoctor	<ul style="list-style-type: none">• On click of 'Add Doctor' button in add-doctor.component.html, this method must be invoked• Should add a doctor to the database by invoking the appropriate method of curewell.service.ts• On successful addition of doctor details<ul style="list-style-type: none">▪ Store the response in the variable 'status'▪ Set the variable 'showDiv' appropriately▪ Set the variable 'msg' to 'Doctor successfully added'• In case of any error<ul style="list-style-type: none">▪ Store the response in the variable 'errorAddMsg'▪ Set the variable 'msg' to 'Some error occurred'• On completion, display the message 'Add doctor completed' on the console

Component Implementation of update-doctor

- The following methods in 'update-doctor.component.ts' need to be implemented

Method Name	Description
ngOnInit	<ul style="list-style-type: none">• Initialize the variables 'doctorId' and 'doctorName' with values received from url route parameters
editDoctorDetails	<ul style="list-style-type: none">• On click of 'Save' button in update-doctor.component.html, this method must be invoked• Should edit the details of a doctor by invoking the appropriate method of curewell.service.ts• If the update is successful,<ul style="list-style-type: none">▪ Store the response in the variable 'status'▪ If the response is true, display the message 'Doctor's name updated successfully!' using alert box. Navigate to ViewDoctorComponent.▪ Else display 'Doctor's name not updated' using alert box. Navigate to ViewDoctorComponent.• In case of any error<ul style="list-style-type: none">▪ Store the response in the variable 'errorMsg'▪ Display the message 'Some error occurred' using alert box▪ Navigate to ViewDoctorComponent.• On completion, display the message 'Updated doctor details successfully.' in the console

Component Implementation of view-specialization

- The following methods in 'view-specialization.component.ts' need to be implemented

Method Name	Description
ngOnInit	<ul style="list-style-type: none">• Invoke the getSpecialization() method of the ViewSpecialization component.
getSpecialization	<ul style="list-style-type: none">• Invoke the appropriate method of curewell.service.ts to fetch the details of all the specializations• In case of successful execution<ul style="list-style-type: none">▪ Store the response in the variable 'specializationList'• In case of exception<ul style="list-style-type: none">▪ Set the variable 'specializationList' to null▪ Store the response in the variable 'errorMsg'▪ Set the variable 'showMsgDiv' appropriately• On completion, display the message 'Specialization Fetched Successfully' in the console

Component Implementation of view-todays-surgery

- The following methods in 'view-todays-surgery.component.ts' need to be implemented

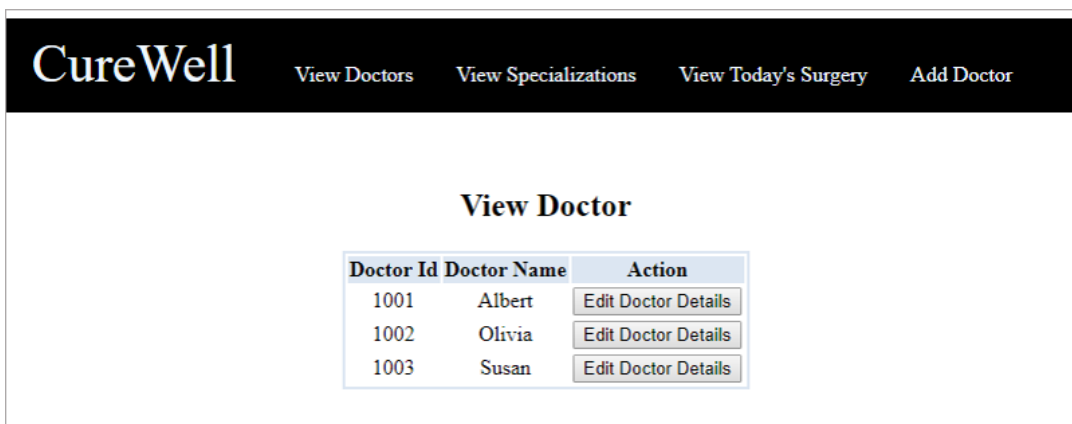
Method Name	Description
ngOnInit	<ul style="list-style-type: none">• Invoke the getTodaySurgery() method of the ViewTodaysSurgery component.
getTodaySurgery	<ul style="list-style-type: none">• Invoke the appropriate method of curewell.service.ts to fetch the details of all the surgeries scheduled for today• In case of successful execution<ul style="list-style-type: none">▪ Store the response in the variable 'surgeryList'• In case of any error<ul style="list-style-type: none">▪ Set the variable 'surgeryList' to null▪ Store the response in the variable 'errorMsg'• On completion, display the message 'Today's Surgery Fetched Successfully' in the console
editSurgery	<ul style="list-style-type: none">• On click of 'Edit Surgery Time' button in view-todays-surgery.component.html, this method must be invoked by passing the required parameters• Navigate to the UpdateSurgeryComponent by passing the required parameters

Component Implementation of update-surgery

- The following methods in 'update-surgery.component.ts' need to be implemented

Method Name	Description
ngOnInit	<ul style="list-style-type: none">• Initialize the variables 'surgeryId', 'surgeryCategory', 'surgeryDate', 'startTime', 'endTime' and 'doctorId' with values received from url route parameters
editSurgery	<ul style="list-style-type: none">• On click of 'Update' button in update-surgery.component.html, this method must be invoked• Should edit the time of a surgery by invoking the appropriate method of curewell.service.ts<ul style="list-style-type: none">• If the update is successful<ul style="list-style-type: none">▪ Store the response in the variable 'status'▪ If the response is true, display 'Surgery details updated successfully!' using an alert box. Navigate to ViewTodaysSurgeryComponent▪ Else display 'Surgery details not updated' using an alert box. Navigate to ViewTodaysSurgeryComponent• In case of an error<ul style="list-style-type: none">▪ Store the response in 'errorMsg'▪ Display the message 'Some error occurred' using an alert box▪ Navigate to ViewTodaysSurgeryComponent• On completion, display 'Updated surgery details successfully' in the console

Implementation of view-doctors.component.html

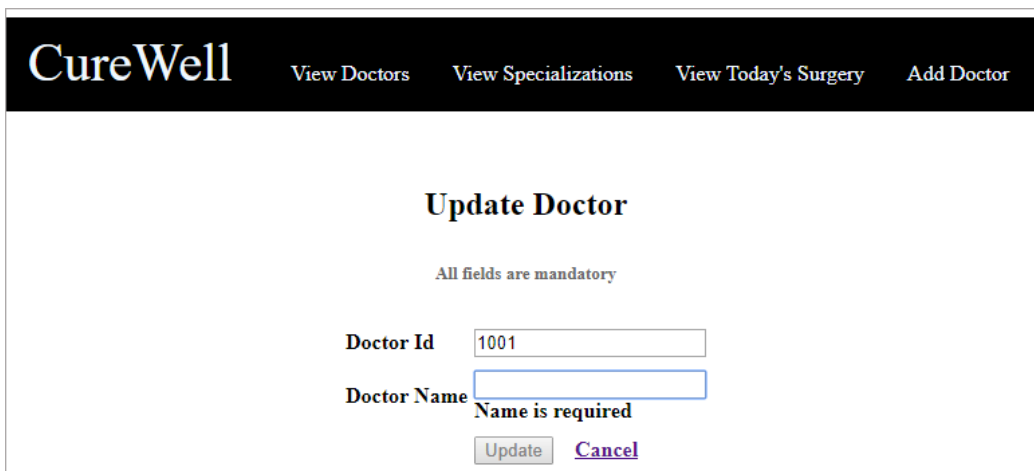


Doctor Id	Doctor Name	Action
1001	Albert	<button>Edit Doctor Details</button>
1002	Olivia	<button>Edit Doctor Details</button>
1003	Susan	<button>Edit Doctor Details</button>

Figure 1-All the doctor details should be populated as shown

Implementation of update-doctor.component.html

- On clicking the Edit Doctor Details button in the ViewDoctors, the Update Doctor page should be displayed by populating the respective doctor's name.
- The Update Doctor view should have a required field validation for Doctor name. The validation message "Name is required" must appear when the text box is touched and dirty.



Update Doctor

All fields are mandatory

Doctor Id

Doctor Name

Name is required

Update [Cancel](#)

Figure 3

Implementation of add-doctor.component.html

- The View Doctor view should have a required field validation for Doctor name.
- The validation message "Name is required" must appear when the text box is touched and dirty.

CureWell
View Doctors
View Specializations
View Today's Surgery
Add Doctor

Add a new Doctor

Doctor Name

[Cancel](#)

Figure 4

On successful addition, the message must be displayed as follows:

CureWell
View Doctors
View Specializations
View Today's Surgery
Add Doctor

Add a new Doctor

Doctor Name

[Cancel](#)

Doctor successfully added

Implementation of view-specializations.component.html

CureWell
View Doctors
View Specializations
View Today's Surgery
Add Doctor

View Specialization

Specialization Code	Specialization Name	Action
ANE	Anesthesiologist	<input type="button" value="View Doctors"/>
CAR	Cardiologist	<input type="button" value="View Doctors"/>
GYN	Gynecologist	<input type="button" value="View Doctors"/>

Figure 5

Implementation of view-todays-surgery.component.html

CureWell
View Doctors
View Specializations
View Today's Surgery
Add Doctor

View Today's Surgery

Surgery Id	Doctor Id	Surgery Date	End Time	Start Time	Surgery Category	Action
5000	1001	01/01/2011	19	17	ANE	<input type="button" value="Edit Surgery Time"/>
5001	1002	01/01/2015	10	15	CAR	<input type="button" value="Edit Surgery Time"/>

- Each surgery detail should have an Edit Surgery Time button which will allow the hospital admin to update the time of the surgery

Implementation of update-surgery.component.html

- The DoctorId, SurgeryDate, and SurgeryCategory must be read-only fields. StartTime and EndTime are required fields.
- The error messages "Start time is required" and "End time is required" must appear when the text boxes are touched and dirty.
- The StartTime must be less than the EndTime.
- Use 24-hour format.

CureWell [View Doctors](#) [View Specializations](#) [View Today's Surgery](#) [Add Doctor](#)

Update Surgery

Surgery Id	<input type="text" value="5000"/>
Doctor Id	<input type="text" value="1001"/>
Surgery Date	<input type="text" value="2011-01-01T00:00:00"/>
Start Time(24 Hours Format)	<input type="text" value="17"/>
End Time(24 Hours Format)	<input type="text" value="19"/>
Surgery Category	<input type="text" value="ANE"/>

[Cancel](#)

CureWell [View Doctors](#) [View Specializations](#) [View Today's Surgery](#) [Add Doctor](#)

Update Surgery

Surgery Id	<input type="text" value="5000"/>
Doctor Id	<input type="text" value="1001"/>
Surgery Date	<input type="text" value="2011-01-01T00:00:00"/>
Start Time(24 Hours Format)	<input type="text"/> Start Time is required.
End Time(24 Hours Format)	<input type="text"/> End Time is required.
Surgery Category	<input type="text" value="ANE"/>

[Cancel](#)