# Project Report – Project Team 1

## Project Team

**Project Lead:**  Darya Lapshina (#300319039)

**Project Team Members:** Thi Hong Gam Tran (#300284929), Gahyun Lee (#300289068)

## Project Description.

The Medical Centre Management application can be used by medical centers, practitioners and their customers. It allows registered patients to find practitioners based on their desired search criteria, book appointments, check their account balance, and make payments on their outstanding charges. Such application improves patients’ access to practitioners and gives practitioners the opportunity to set their schedules, review and provide comments on each of their bookings. It also allows administration staff to back up from and restore data to the database. Application is created considering Front Desk staff members as the main actor, who are facilitating all features and functions.

Deliverables of this project include a Project Proposal document (delivered earlier), an SQL Server Database Project with the necessary sample data, and a Windows Forms Application created with C# .NET Entity Framework.

**Core features/requirements of the application include:**

* Register (New patients who would like to use the application can create a user account through a staff member.)
* Update Patient Information (Patient can update their personal information)
* Search (Registered patients can use the search feature to find practitioners based on their desired criteria.)
* Book an Appointment (Patients can use this feature to book appointments to see a practitioner of their choosing and pick the services they desire.)
* Cancel an Existing Appointment (Patients and Practitioners can cancel future bookings. Refunds are issued if the booking has been paid for.)
* Payment Management (Patients have access to their bill information, pay their outstanding fees, and check their payment history.)
* Practitioner Registration (Front desk staff can register new practitioners)
* Update Practitioner Information (Practitioners can update their personal information)
* Schedule Management (Practitioners can use this feature to manage their availability for bookings.)
* Database backup and restore (Staff members can use this feature to backup and restore the database when needed.)

## Process Model.

**Main User Roles/Actors within the application:** Staff Members (Front Desk), Patients, Practitioners.

**User Processes:**

* Patients can update their information, search practitioners, see their availability, and book appointments for certain services. They also have access to their own appointment/payment history, pay their outstanding balance, cancel their future appointments, and read practitioner’s comments on each one of their appointments.
* Practitioners can update their information, set up their availability, see their bookings, update bookings statuses, and provide comments to the patient upon completion of the appointment.
* Staff members can register new users (both patients and practitioners), backup and restore the database.

## Database Model.

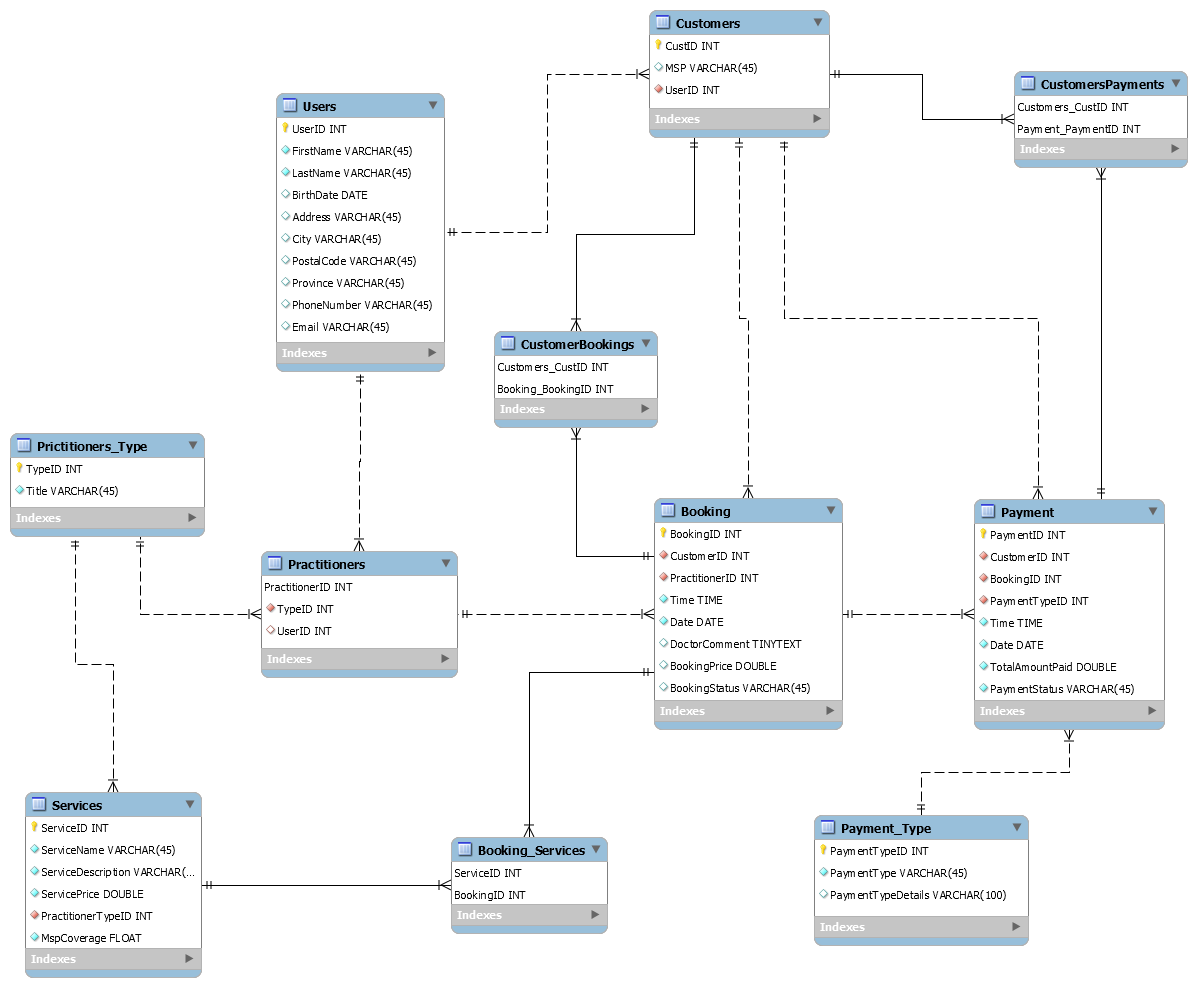
**Main Entities** *(will be implemented as tables in the project):*

* USERS (UserID,FirstName, LastName, BirthDate, Address, City, Province, PhoneNumber, Email)
* CUSTOMERS (CustomerID, *UserID*, MSP)
* PRACTITIONERS (PractitionerID, *TypeID, UserID*)
* PRACTITIONERS\_TYPE (TypeID, title)
* SERVICES (ServiceID, ServiceName, ServiceDescription, ServicePrice, *PractitionersTypeID,* MSPCoverage)
* BOOKINGS (BookingID, *CustomerID*, *DoctorID*, Time, Date, DoctorComment, BookingPrice, BookingStatus)
* PAYMENT (PaymentID, *BookingID*, *CustomerID, PaymentTypeID,* TotalAmountPaid,PaymentStatus, Time, Date)
* PAYMENT\_TYPE (PaymentTypeID, PaymentType, PaymentTypeDetails)

**Associative Entities or intermediate tables due to the many-to-many relationships** *(Will become navigation properties in the project):*

* BOOKING\_SERVICES (*BookingID*, *ServiceID*)
* CUSTOMER\_BOOKINGS (*CustomerID, BookingID*)
* CUSTOMER\_PAYMENTS (*CustomerID, PaymentID*)

**ER Diagram:**



*Figure 1. ER Diagram of the database for the health management business application*

## User Interface

**Number of screens**: 12

1. Main Options Screen (Staff can choose Administrative or Records-related option)
2. All Records Screen (All Patients and Practitioners are displayed. Delete/Add options are available, and a specific record can be selected)
3. Register a new Patient (Staff can input data into fields to register a new customer)
4. Register a new Practitioner (Staff can input data into fields to register a new practitioner)
5. Patient home screen (Patient Bookings/Payments are listed. Other options are available)
6. Patients Search/Booking (Patients can search practitioners by inputting different criteria and book appointments)
7. Patients Making Payment (This screen will allow Patients to make payment)
8. Patient’s Update Screen (This screen is for update of Patient’s information)
9. Practitioner home screen (Practitioner Bookings are displayed. Other options are available)
10. Practitioner’s Update Screen (This screen is for update of Practitioner’s information)
11. Practitioner schedule (Practitioners can access and update their schedule and availability)
12. Administration form

## Implementation.

The Medical Centre Management Application Solution consists of the 8 following projects:

1. DataTableAccessLayer
2. MedicalCentreCodeFirstFromDB
3. MedicalCentreDB
4. MedicalCentreMainMenu
5. MedicalCentrePatient
6. MedicalCentrePractitioner
7. MedicalCentreUtilities
8. MedicalCentreValidation

**DataTableAccessLayer**

This project contains utility methods inside a SqlDataTableAccessLayer class, that are used to backup the database to an XML file and restore it from an existing XML file.

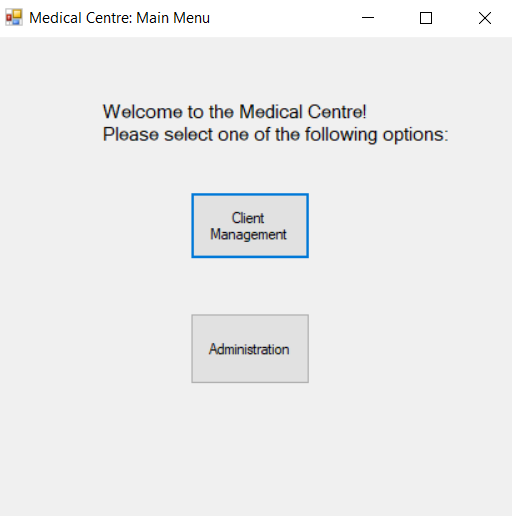
**MedicalCentreCodeFirstFromDB**

This project contains the Entity Data Model (all classes created from the database), static class used to seed the database, and a static class to override ToString() methods of some of the entity classes. It also includes Status constants which include Enums for BookingStatus and PaymentStatus.

**MedicalCentreDB**

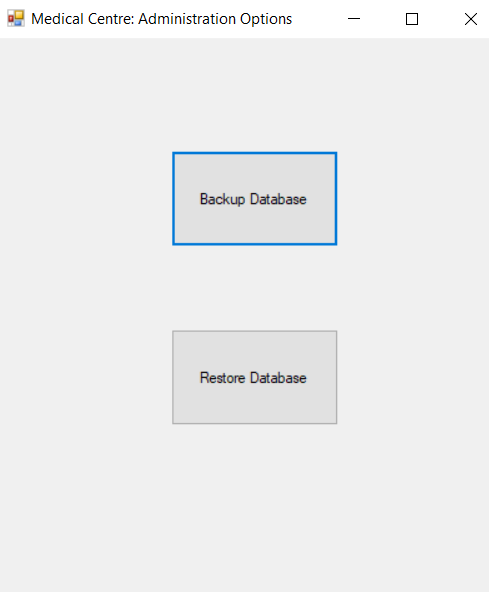
This is the SQL database including all tables used throughout the application. All attributes and field properties are included. This project was used as a base for Entity Data Model creation. This project also contains a publish XML file that can be used to create a database.

**MedicalCentreMainMenu (Windows Forms Project)**This project includes Main Menu options of the application and was implemented using child forms.



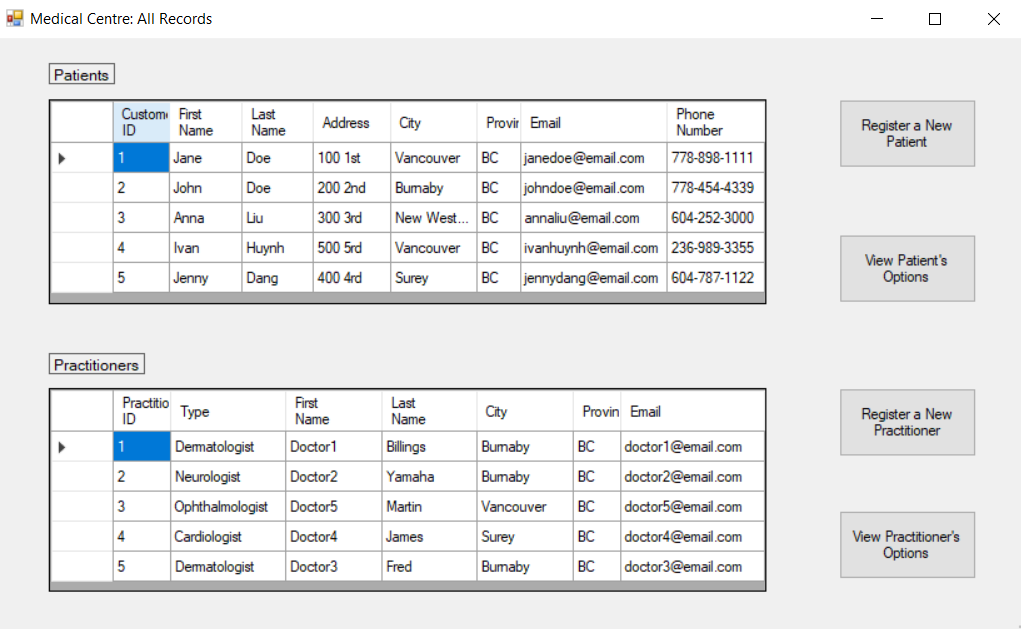
The Main Menu screen presents users with two options: Client Management or Administration. Upon clicking one of those buttons, a related child form is displayed.

*Figure 2. Medical Centre: Main Menu Windows Form from MedicalCentreMainMenu project.*



The Administrative Options child form contains two buttons: Backup database and Restore database. Upon clicking the Backup button, the database will be backed up into an XML file, while the Restore button will read from the XML file back to the database.

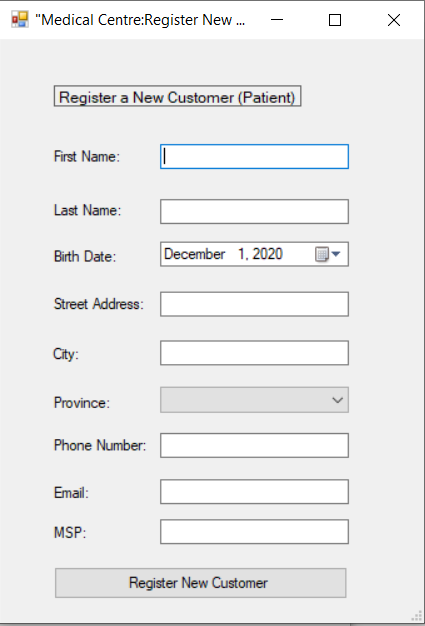
*Figure 3. Medical Centre: Administrative Options Windows Form from MedicalCentreMainMenu project.*

The MedicalCentreAllRecordsForm class displays both all Customer and all Practitioner records. A few buttons are displayed, that load their child forms. Buttons “View Patient’s Options” and “View Practitioner’s Options” require users to select one row in the associated DataGridView prior to proceeding. 

*Figure 4. Medical Centre: All Records Windows Form from MedicalCentreMainMenu project.*

**MedicalCentrePatient (Windows Forms Project)**

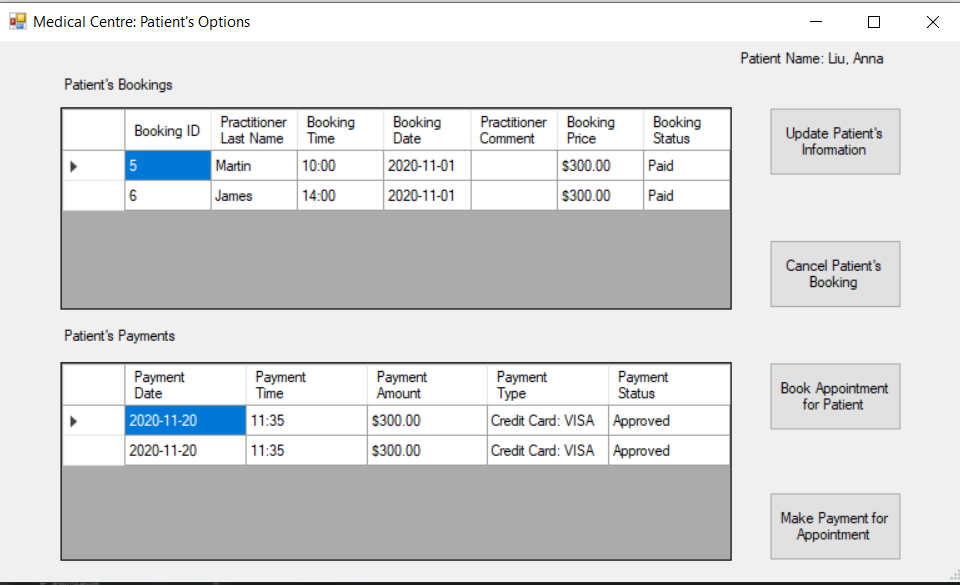
This project includes all forms and functionalities that involve Customers of a Medical Centre.



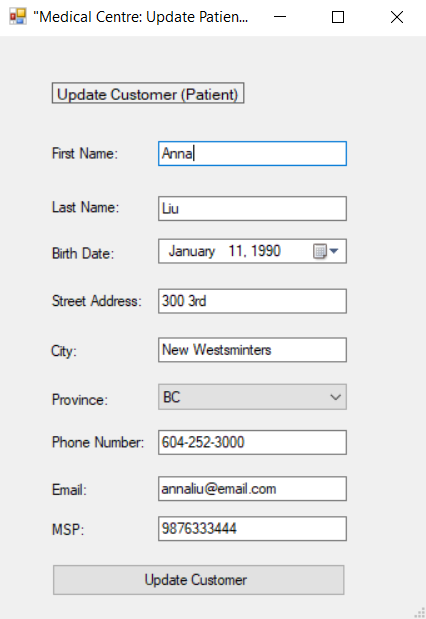
MedicalCentreAddPatient class displays the following form (as seen on Figure 2). Users of the system can input data into the fields and press the button to add a new Customer to the database. User inputs are validated to ensure that all required data is passed in and is of appropriate length and format.

*Figure 5. Medical Centre: Register New Customer Form.*

The MedicalCentrePatientOptionsMainForm class is the home page of every Customer. It displays a particular customer’s bookings and payments. A few more options are available, such as updating their personal information, cancelling their bookings, booking a new appointment, and making payment for an outstanding charge. Cancelling the Booking requires the User to select a Booking to be cancelled and validates its date. Depending on the status of the Booking, payment status is updated if required. Customers have 7 days to complete the payment after the booking date, if no payment is received their account is suspended and no further bookings can be made until the payment is received.



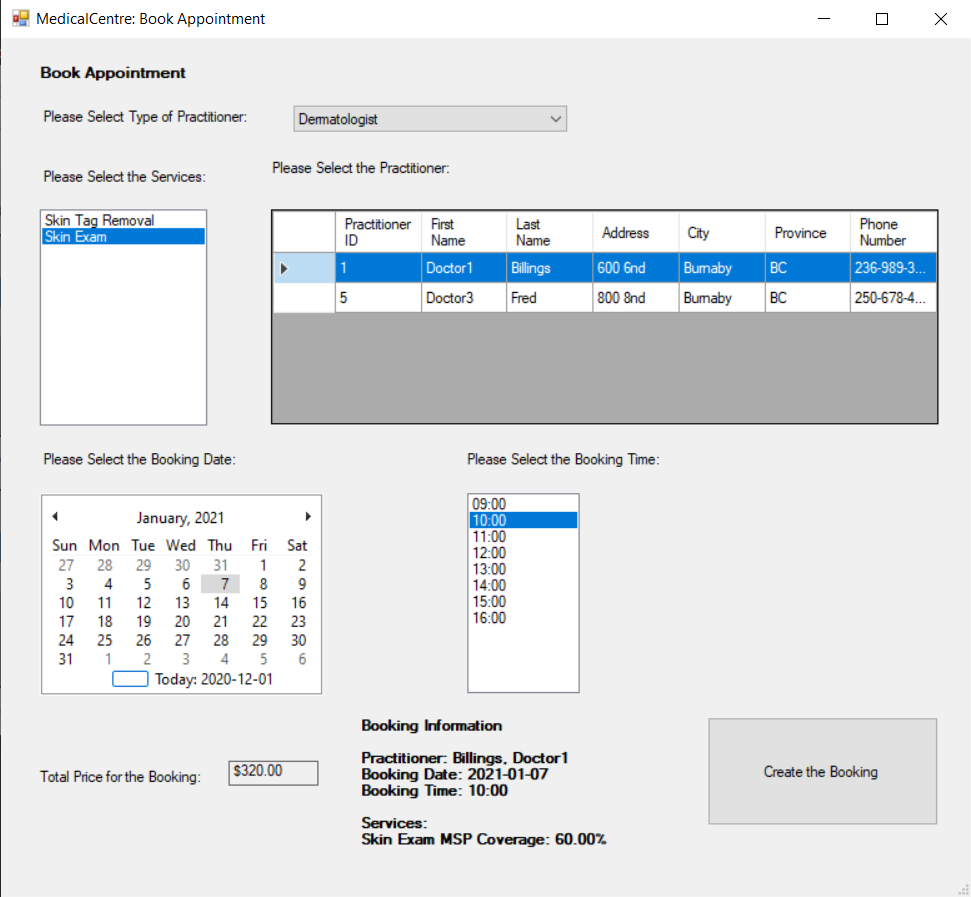
*Figure 6. Medical Centre: Patient’s Options Windows Form from MedicalCentrePatient project.*



MedicalCentreUpdatePatient class displays the following form (as seen on Figure 3). The inputs are prepopulated with Customer’s current data. System users can change the values of fields as desired and update the database upon button click. Prior to the update, all values are validated, and error messages are displayed if newly inputted data is invalid. Changes to MSP update prices for any unpaid future bookings according to the MSP coverage of each service within the booking.

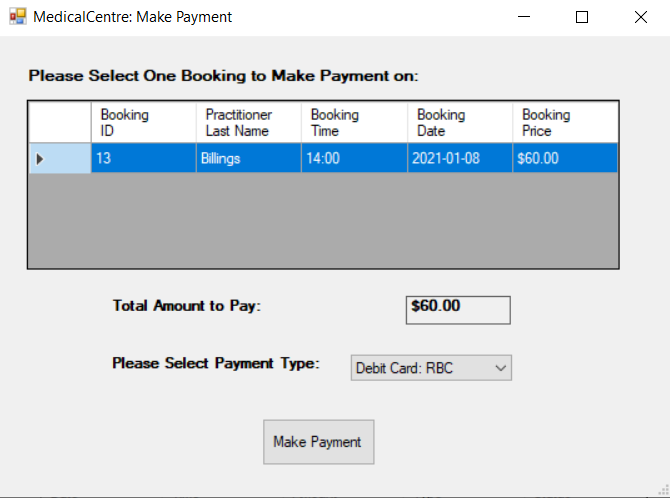
## 

*Figure 7. Medical Centre: Update Patient Windows Form from MedicalCentrePatient project.*



MedicalCentreBookAppointment Class displays a Windows form that allows a Customer to book an appointment. Upon selecting a Type of Practitioner, the lists of services and practitioners are displayed. Users can select multiple services in a booking but only one practitioner from the datagridview. Once Date and Time have been selected, the summary of Booking Information is displayed as well as the total price for the booking. Price is calculated depending on Customer’s MSP coverage. Clicking the button will create the booking and add it to the database upon validating all information.

*Figure 8. Medical Centre: Book Appointment Windows Form from MedicalCentrePatient project.*

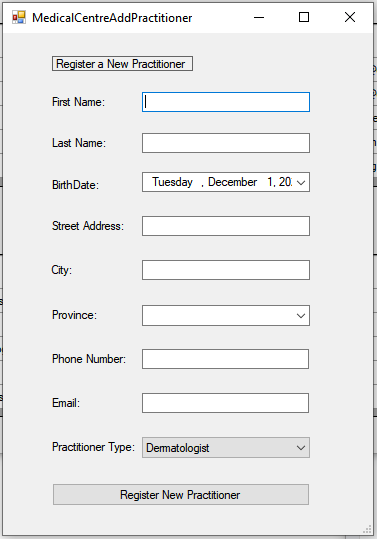


MedicalCentreMakePayment Class displays a Windows form that allows a Customer to pay for their unpaid bookings. Upon selecting one booking from the datagridview, the total amount is updated. Clicking the button will create a Payment, insert the record into the database and update the Booking’s status. Validation is performed prior and error messages are displayed.

*Figure 9. Medical Centre: Make Payment Windows Form from MedicalCentrePatient project.*

**MedicalCentrePractitioner (Windows Forms Project)**

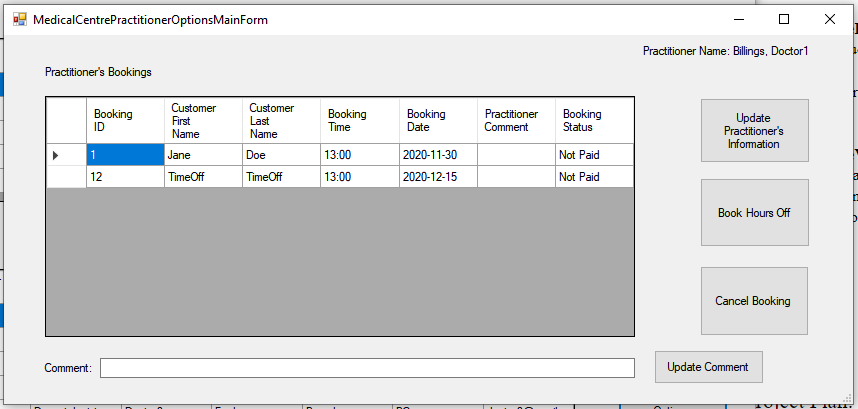
This project includes all forms and functionalities that involve Practitioners of a Medical Centre.



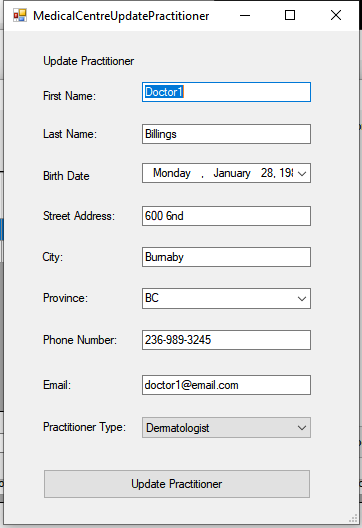
MedicalCentreAddPractitioner Class displays the following form (as seen on Figure 10). Users of the system can input data into the fields and press the button to add a new Practitioner to the database. User inputs are validated to ensure that all required data is passed in and it is of appropriate format.

*Figure 10. Medical Centre: Register New Practitioner Form.*

The MedicalCentrePractitionerOptionsMainForm class is the home page of every Practitioner. It displays a particular practitioner’s bookings. A few more options are available, such as updating their personal information, cancelling their bookings, booking a time off, and updating practitioner’s comments. Cancelling the Booking requires the User to select a Booking to be cancelled and validates its date. Depending on the status of the Booking, payment status is updated if required.



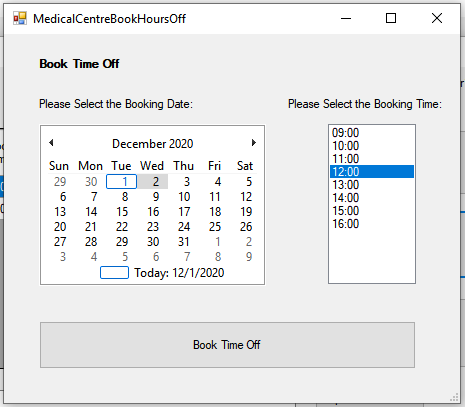
*Figure 11. Medical Centre: Practitioner’s Options Windows Form from MedicalCentrePractitioner project.*



MedicalCentreUpdatePractitioner class displays the following form (as seen on Figure 12). The inputs are prepopulated with Practitioner’s current data. System users can change the values of fields as desired and update the database upon button click. Prior to the update, all values are validated, and error messages are displayed if newly inputted data is invalid.

*Figure 12. Medical Centre: Update Practitioner Windows Form from MedicalCentrePractitioner project.*

MedicalCentreBookHoursOff Class displays a Windows form that allows a Practitioner to book hours off. Once Date and Time have been selected, the summary of Booking Information is displayed. Clicking the button will create the booking and add it to the database upon validating all information.



*Figure 13. Medical Centre: Book Hours Off Windows Form from MedicalCentrePractitioner project.*

**MedicalCentreUtilities**

This project contains utility methods inside a static Controller class. This class is generic and is used throughout the project to perform CRUD operations on the database. This project also includes a BaseFormsMethods static class, containing generic form setup methods used throughout the application.

**MedicalCentreValidation**

This project contains all validation methods, checking properties of objects, used throughout the application. It consists of 5 static classes: BookingValidation class, CustomerValidation class, PaymentValidation class, PractitionerValidation class, and UserValidation class. UserValidation is performed using Regex Class to ensure that fields are of correct length and format.

## Application Configuration and Running Instructions.

**Configuration Instructions:**

1. Extract the folder ProjectTeam01TermProject.zip.
2. Go inside the ProjectTeam01MedicalCentreManagement Folder.
3. Open the Solution in Visual Studio (ProjectTeam01MedicalCentreManagement.sln)
4. Run the MedicalCentre.publish.xml file found inside the MedicalCentreDB project to publish the Database.
5. Set the Startup Project to MedicalCentreMainMenu (VS keeps resetting it to the MedicalCentreDB).
6. Run the Application.

**Running Instructions:**

1. Upon loading the project- Welcome Screen is displayed.
2. Administration button leads to a screen that performs backup and restore of the Database.
3. Client Management button leads to a screen where all Customers and Practitioners are displayed.
4. To add a Customer/Practitioner, press the appropriate Register button. Please note, that input validation includes the following fields and rules: “First Name”, “Last Name” (not empty and contain letters), “Email” (not empty and of format “email@email.com”) and “Phone Number” (not empty and few formats are accepted, such as “111-111-1111”, “(111)-111-1111” and “111 111 1111”).
5. To view options of a specific Customer/Practitioner, select the row containing that user in the appropriate DataGridView and press the View Options button.

**Running Instructions (Patient Features)**

1. Please note, that if a Patient has any unpaid bookings from 7 days ago or more (example John Doe with CustomerID 1), an error message is displayed, and Book Appointment feature is disabled until payment is received.
2. While in the Patient’s Main Options Menu, a few features can be accessed, such as: “Update Customer”- which would open a window similar to Registration (Please note, same validation applies), “Cancel Booking”- requires selecting a booking from the DataGridView (Please note, past bookings cannot be cancelled, and future paid bookings will be refunded).
3. To book an appointment, press the “Book an Appointment” button from the Patient’s Main Menu, which would open a child form. Selecting a type of the practitioner, will change the list of available services and doctors. Select one or many services from the ListBox and only 1 Practitioner from the DataGridView. Select desired date of the Booking (no past bookings are permitted), the ListBox of times will be populated based on your selected Practitioner availability (Please note, if choosing today as the Booking date- available times will be adjusted based on current time). Select the time slot from the ListBox, double check your booking price and summary (price calculated based on MSP coverage of each service and whether the specific customer has MSP) and press “Book appointment”. This new booking will be displayed in the Patient’s bookings DataGridView.
4. To make a payment, press the “Make a Payment” button from the Patient’s Main Menu, which would open a child from (if that Customer has any unpaid bookings). Select a booking from the DataGridView- upon selection the price is shown. Select the payment method from the ComboBox and press “Make Payment”. New payment will be displayed in the Patient’s payments DataGridView and booking status will be updated.

**Running Instructions (Practitioner Features)**

1. While in the Practitioner’s Main Options Menu, a few features are available: “Update Practitioner”- which would open a window similar to Registration (Please note, same validation applies), “Cancel Booking”- requires selecting a booking from the DataGridView (Please note, past bookings cannot be cancelled, and future paid bookings will be refunded). DataGridView cannot be edited but can be sorted by any desired column.
2. Update Comment feature requires a selection of a booking from the DataGridView and an input of the comment in the TextBox, found below. An error message is displayed if a comment is blank or no booking is selected.
3. “Book Hours Off” button leads to a child form, where the Practitioner can select times to not be available for bookings. Selecting a date (Please note, past dates are not permitted) populates a ListBox with times that can be booked for time-off (not already booked by customers). Please note, if choosing today as the Booking date- available times will be adjusted based on current time. Upon selecting both time and date, press the “Book Time off” button to finish the request. An error message is displayed if information is invalid.

## Project Tasks.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Task | Person | Time start | Timeframe | Progress |
| Administration User Interface *(Form Design - 1 Screen)* | Thi Hong Gam | 2020-11-05 | 2 weeks | Date Completed:  (2020-11-19) |
| Main Menu User Interface *(Forms Design - 2 Screens)* | Darya | 2020-11-05 | 1 week | Date Completed:  (2020-11-12) |
| Patient User Interface *(Forms Design - 5 Screens)* | Darya | 2020-11-05 | 2 weeks | Date Completed:  (2020-11-19) |
| Practitioner User Interface  *(Forms Design - 4 Screens)* | Gahyun | 2020-11-05 | 2 weeks | Date Completed:  (2020-11-19) |
| Database creation *(Creating SQL tables)* | Thi Hong Gam | 2020-11-05 | 1 week | Date Completed:  (2020-11-12) |
| User Input Validations *(MedicalCentreValidation project implementation)* | Gahyun | 2020-11-12 | 3 weeks | Date Completed:  (2020-12-01) |
| Regex Expressions Implementation | Darya | 2020-11-30 | 1 week | Date Completed:  (2020-12-02) |
| Admin Functionality Features  *(Backup and Restore using DataTableAccessLayer Project)* | Darya | 2020-11-12 | 3 weeks | Date Completed:  (2020-12-01) |
| Patient Functionality Features:  *Add/Update* *Book Appointment.*  *Cancel Booking. Make Payment.*  *Input Validation.* | Darya | 2020-11-12 | 3 weeks | Date Completed:  (2020-12-02) |
| Practitioner Functionality Features: *Add/Update Book Time-off.*  *Cancel Booking. Leave Comment.*  *Input Validation.* | Gahyun | 2020-11-12 | 3 weeks | Date Completed:  (2020-12-02) |
| Utilities and Constants Creation: *(ToString() overrides, Enums, BaseFormMethods)* | Darya/Gahyun | 2020-11-12 | 2 weeks | Date Completed:  (2020-11-22) |
| Documentation *(Project Report Creation)* | Darya/Thi Hong Gam/Gahyun | 2020-11-26 | 1 week | Date Completed:  (2020-12-05) |
| Integration *(Merging all 8 projects to create a complete application)* | Darya/Thi Hong Gam/Gahyun | 2020-11-23 | 1 week | Date Completed:  (2020-12-01) |
| Testing *(Ongoing throughout the project)* | Darya/Thi Hong Gam/Gahyun | 2020-11-12 | 4 weeks | Date Completed:  (2020-12-05) |