CDI

NL Hospital

Integrated Project – P2

Bruna Maciel (635-203263)
6-15-2020

Contents

Project Description

This project simulates a simplified hospital system. This implementation is an adaptation/continuation of the student files project provided by the CDI for this assignment.

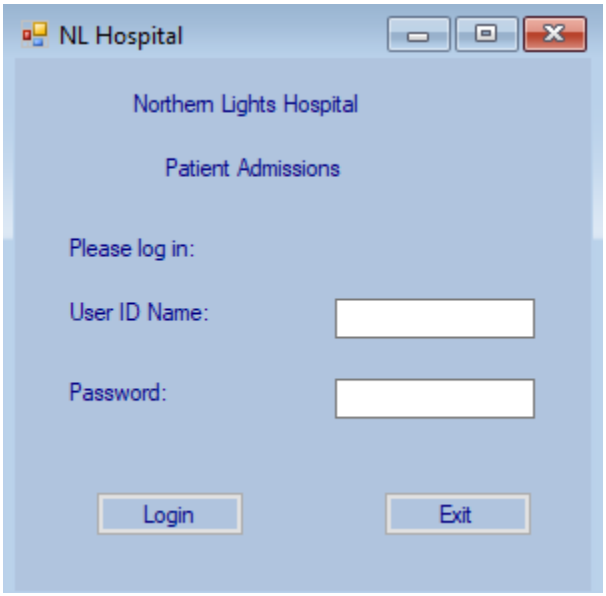
The initial project used an Access database and as part of the assignment for this project it was changed to SQL Server database. With that said, it's necessary create the SQL database before running this project.

Key Functions

- Login:
  Username and passwords are stored in database. Each login is linked with an employee and each employee type has access to different parts of the system.
  The login allows the system functions based on the employee type linked to the username.
- Manage Employees:
  Add, edit, find employees.
  Only employees from administration can see this functionality.
- Bill Patient:
  Bill patients for extra amenities after they are discharged.
  Only employees from administration can see this functionality.
- Discharge Patients:
  Discharge the patient and frees the bed he/she was using.
  Only doctors can see this functionality.
- Surgery Report:
  Report with all scheduled surgeries. It shows only the admissions with scheduled surgeries, if there is none, the report shown its empty. In the populated database, the last surgery added is in '06-08-2020'.
  Only doctors and nurses can see this functionality.
- Manage Patients
  Add, edit, find patients.
  Only employees from admission can see this functionality.
- Hospitalized Patients
  Report with all patients currently hospitalized
  Only employees from admission and nurses can see this functionality.
- Manage Admissions
  Add, edit, find admissions.
  Only employees from admission can see this functionality.
- Admissions Report
  Report with all the hospital's admissions.
  Only employees from admission and nurses can see this functionality.

Application Screens

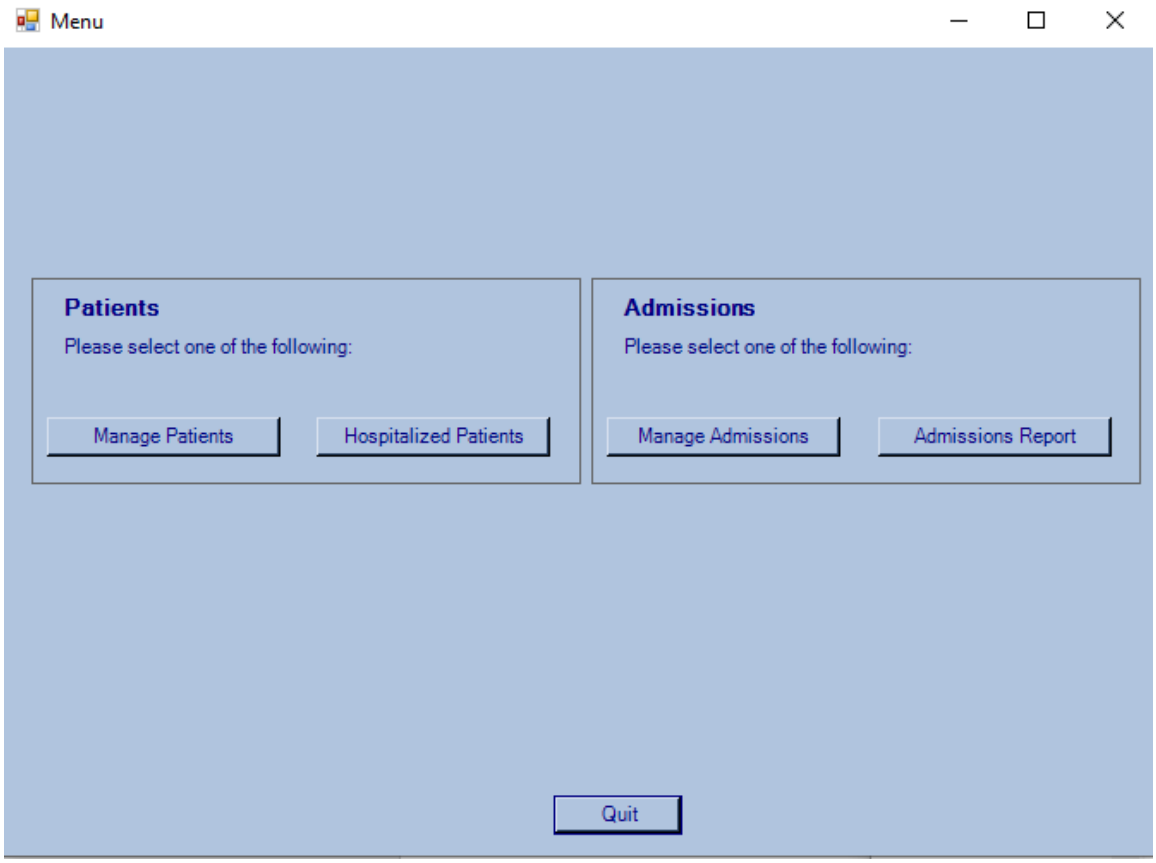
- Login



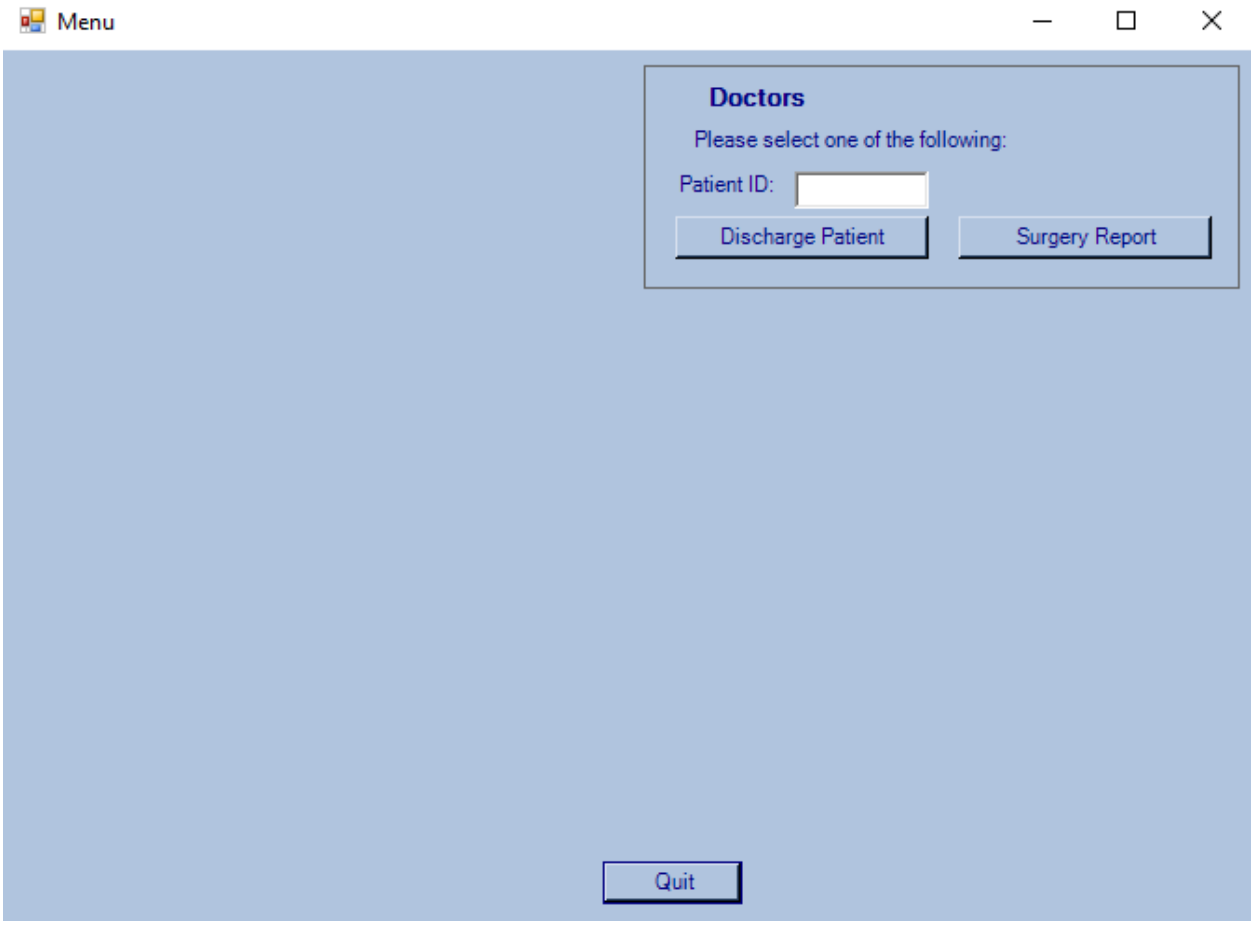
- Menu

The panels visibility changes accordingly with the user login.

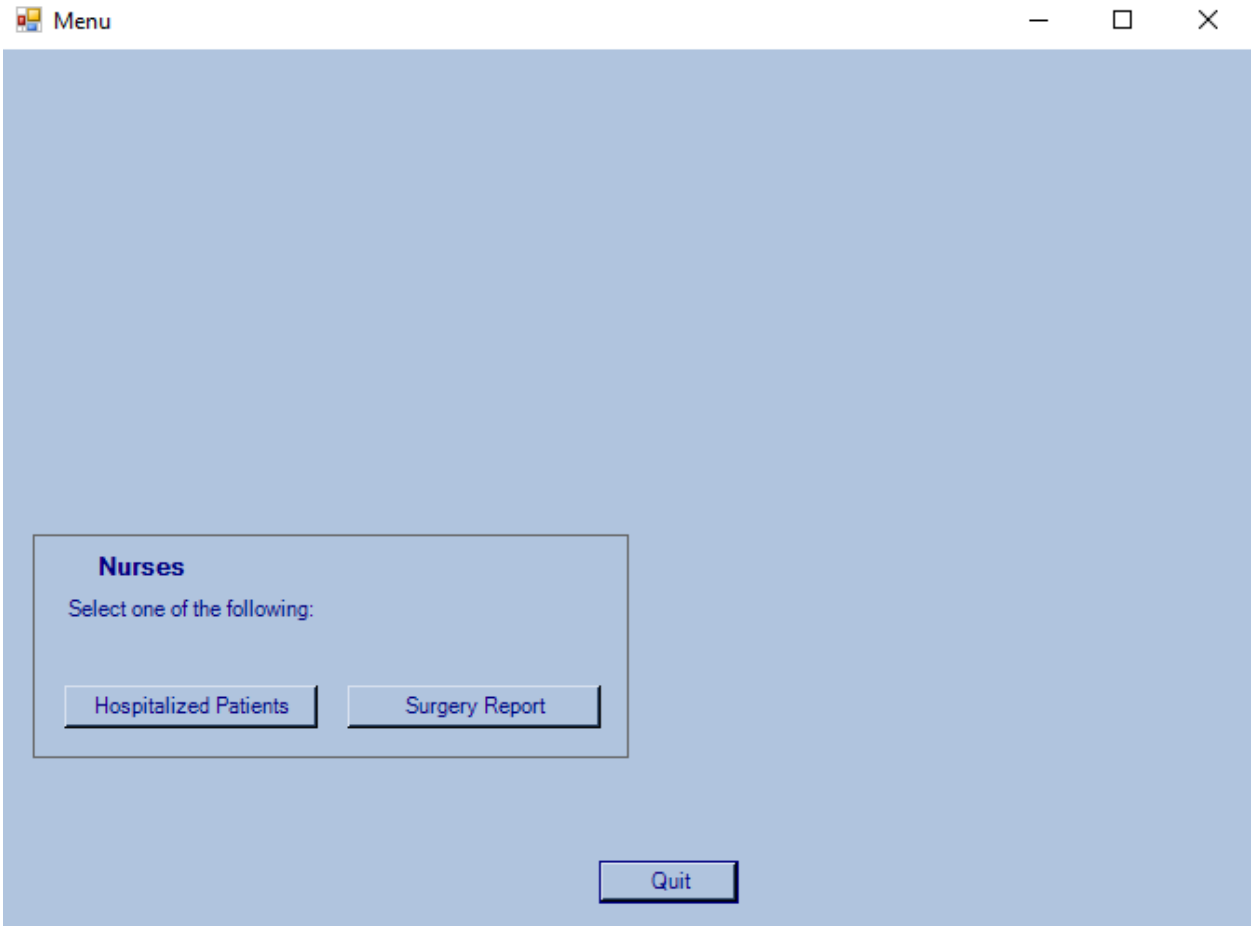
- ❖ Admissions employee:
  - Login: *bourneJ* , Password: *1234*



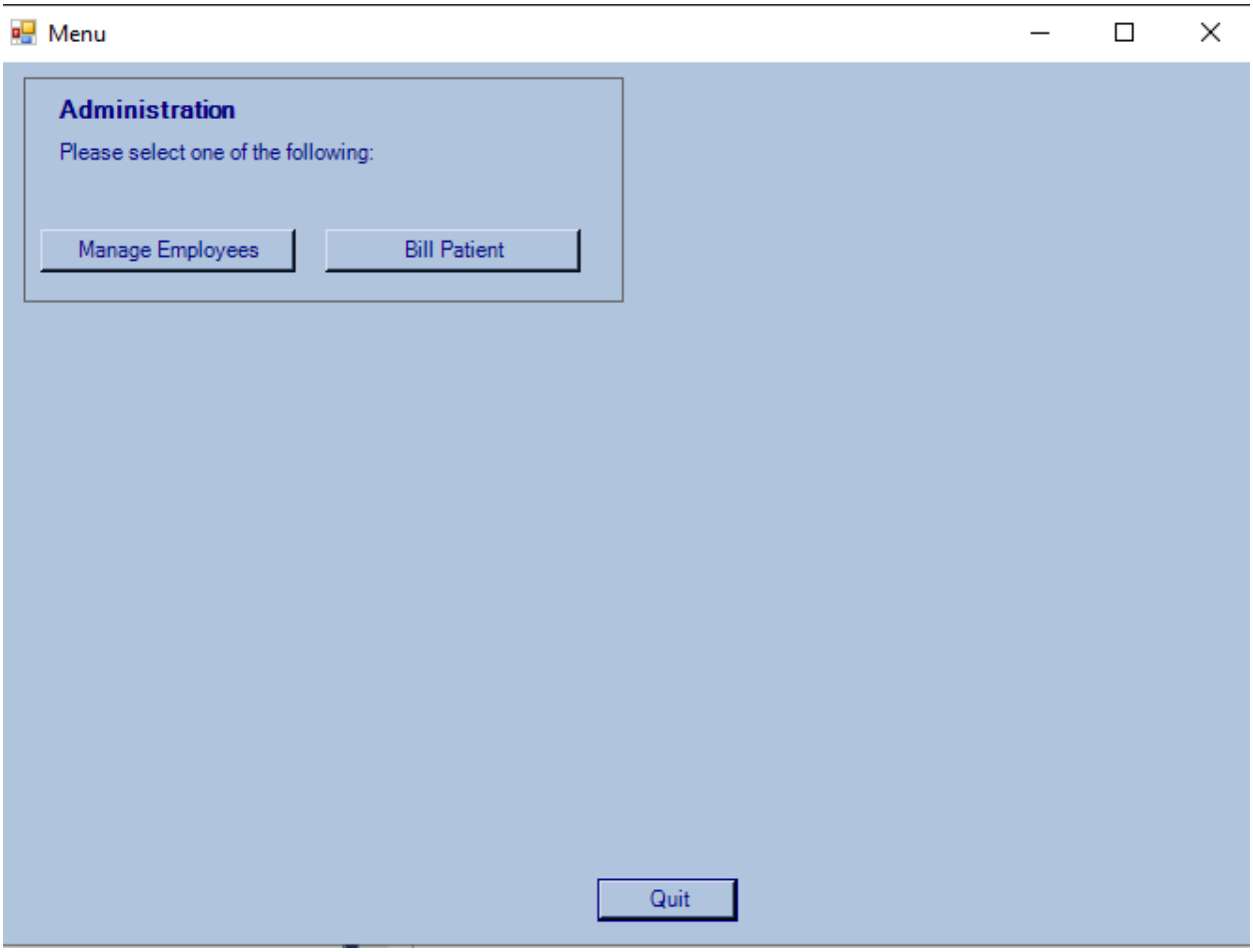
- Doctor employee:
  - Login: *connorS* , Password: *1234*
  - Login: *salanderL* , Password: *1234*
  - Login: *huntE* , Password: *1234*



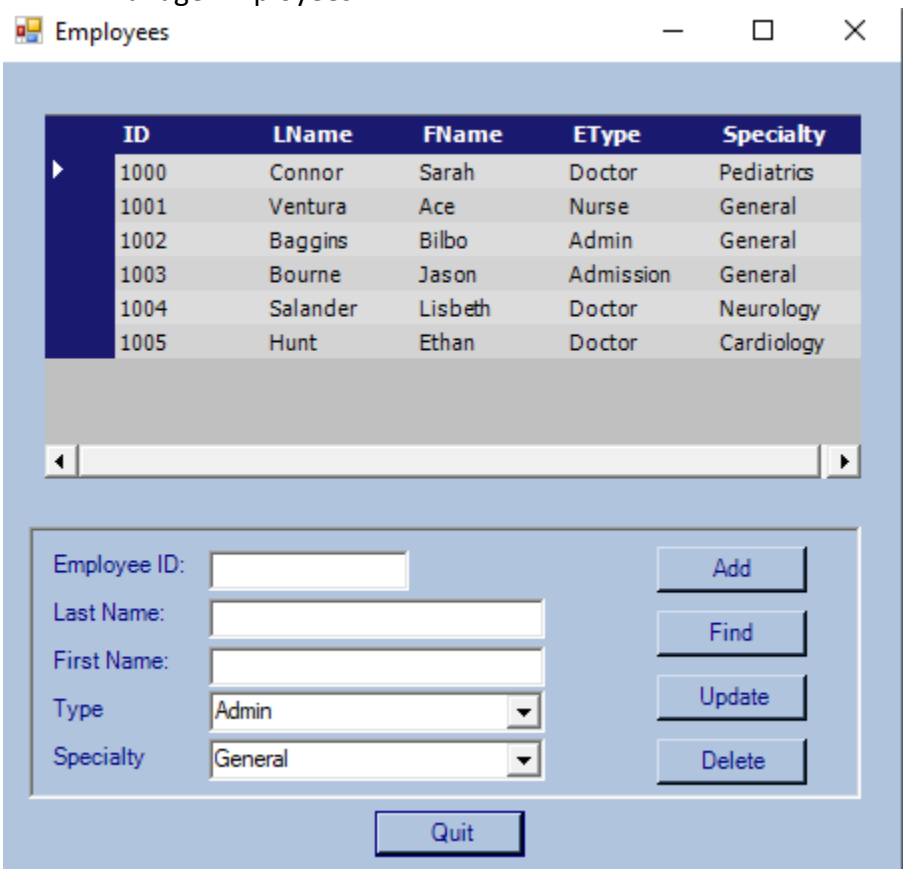
- Nurse employee
  - Login: *venturaA*, Password: *1234*



- Administration employee
  - Login: *baggingsB* ,Password: *1234*

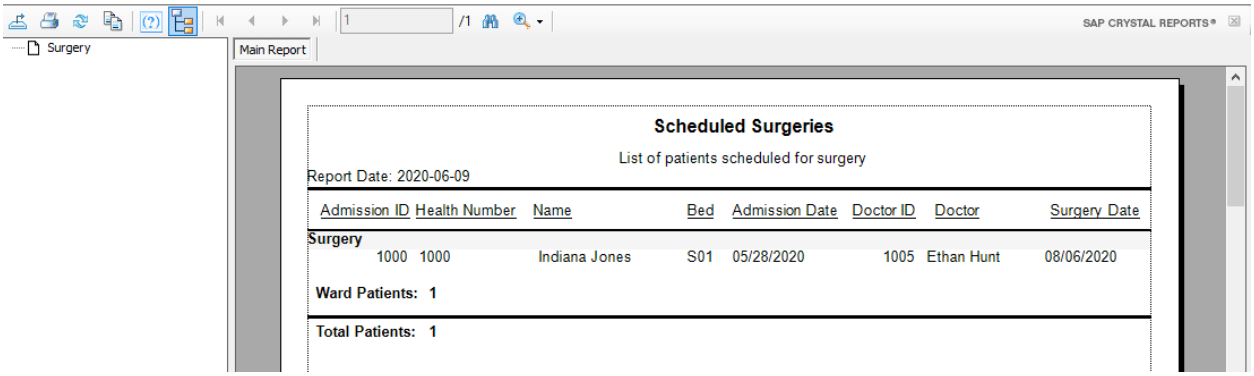


- Manage Employees:

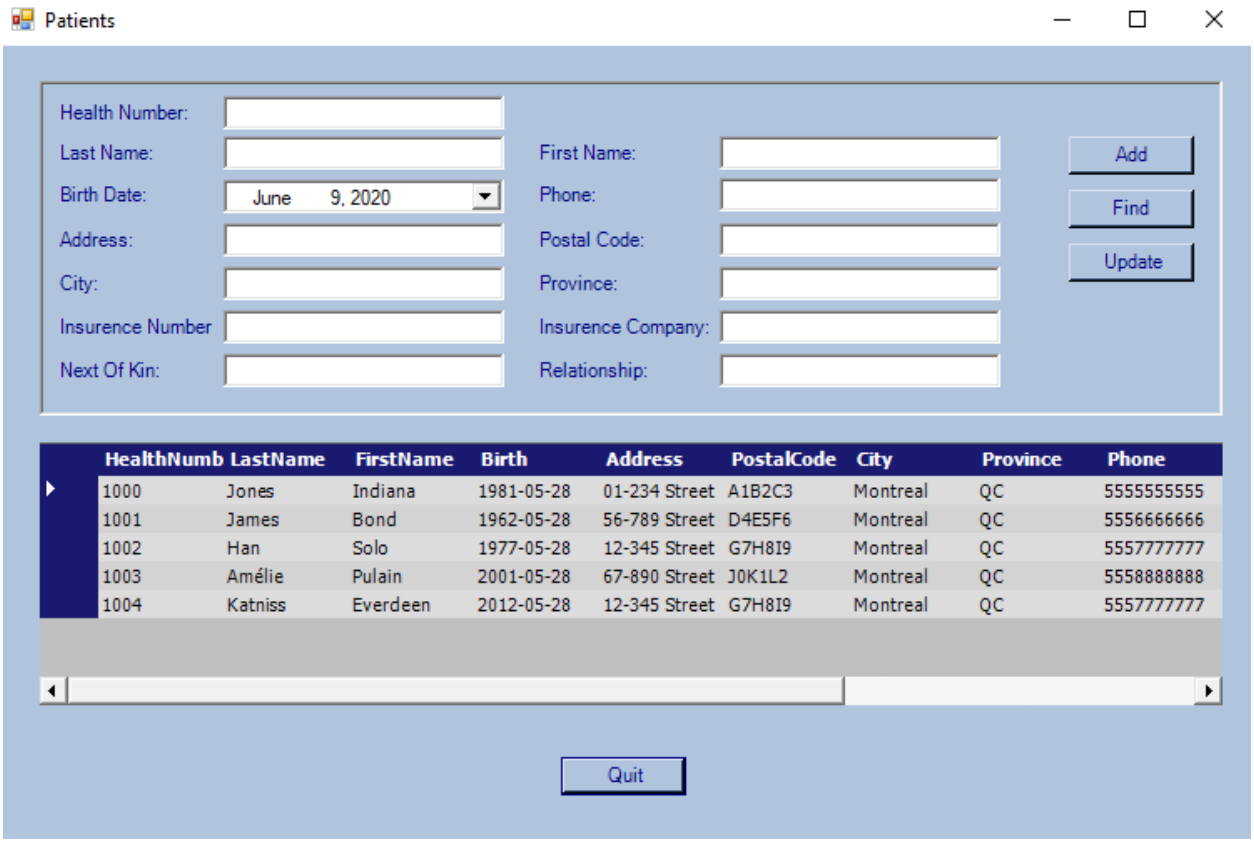


- Bill Patient:

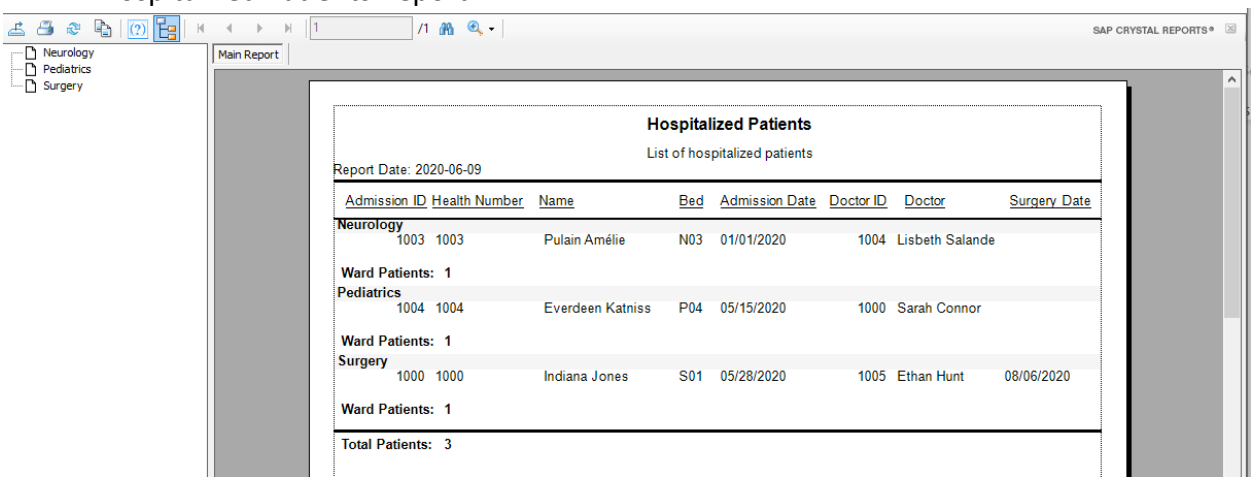
- Surgery Report



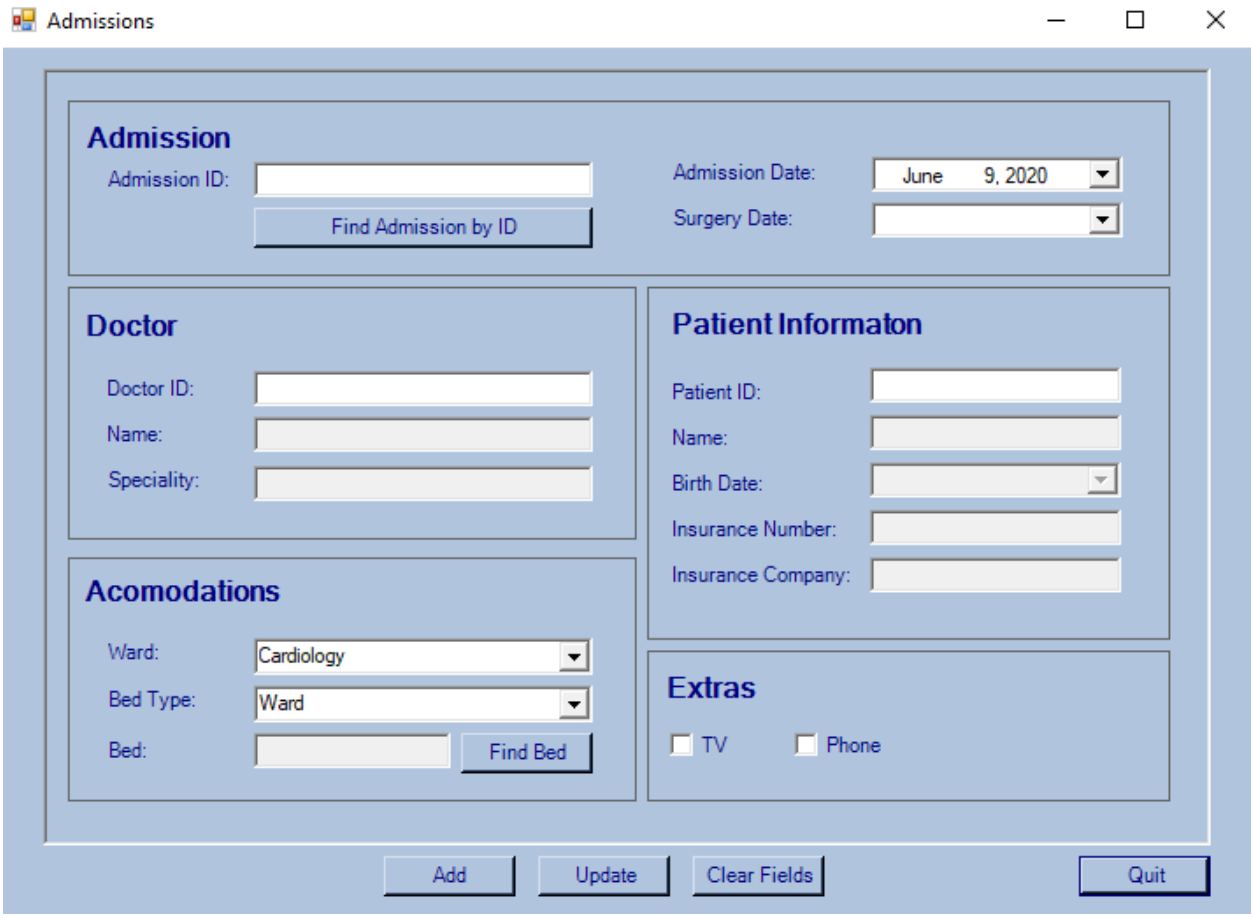
- Manage Patients



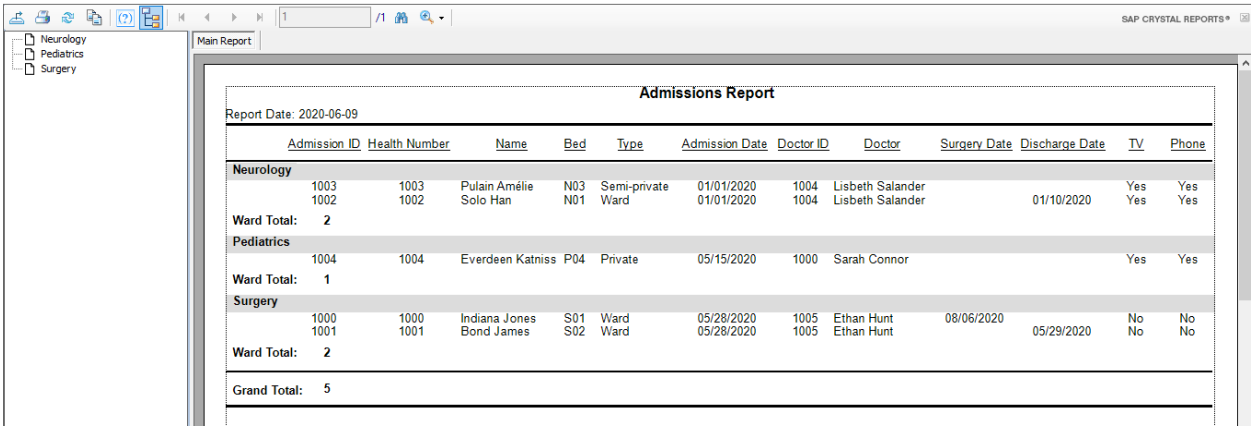
- Hospitalized Patients Report



- Manage Admissions



- Admissions Report



Admissions Report

Report Date: 2020-06-09

	Admission ID	Health Number	Name	Bed	Type	Admission Date	Doctor ID	Doctor	Surgery Date	Discharge Date	TV	Phone
Neurology												
	1003	1003	Pulain Amélie	N03	Semi-private	01/01/2020	1004	Lisbeth Salander			Yes	Yes
	1002	1002	Solo Han	N01	Ward	01/01/2020	1004	Lisbeth Salander		01/10/2020	Yes	Yes
Ward Total:	**2**											
Pediatrics												
	1004	1004	Everdeen Katniss	P04	Private	05/15/2020	1000	Sarah Connor			Yes	Yes
Ward Total:	**1**											
Surgery												
	1000	1000	Indiana Jones	S01	Ward	05/28/2020	1005	Ethan Hunt	08/06/2020		No	No
	1001	1001	Bond James	S02	Ward	05/28/2020	1005	Ethan Hunt		05/29/2020	No	No
Ward Total:	**2**											
Grand Total:	**5**											

Assignment Steps

Database Design

Access Tables	SQL Tables	What changed
AdmissionRecords	Admissions	1-surgery y/n field was deleted 2- Incorporated Access Extras table fields 3-Add doctor field
Beds	Beds	
Doctors	Employees	1-Keeps all employees records not only doctors 2-New field to indicate the employee type
Extra_Rates	ExtraRates	
Extras		1-Table deleted; the fields are now part of the Admission's table
Login	Logins	1-New field that link the login with an employee
Patients	Patients	1-Doctor field removed
-	BedType	1-New table that keeps the possible types of bed
-	EmployeeType	1-New table that keeps the types of employee
-	Specialties	1-New table that keeps the employee's specialties
-	Wards	1-New table that keeps the hospital wards

Why change

Admissions table now have all information necessary about the patient admission including the extras, this way it's easier to bill after the discharge. The doctor field was added to this table and deleted from the Patients table since a patient can have multiple admissions for different reasons and so have different doctors. There is no need for 2 fields to inform if there is a surgery scheduled, if the surgery date field is filled the patient has a surgery scheduled.

Doctor table now is called Employees and keeps all the employee's information not only doctors this way it's easier to associate the login with each employee and keep the employees' information's (that were not keep in the access database).

New supporting tables were created to keep the information's that rarely change (i.e. hospital wards, bed types, employees' types and specialties.

Business Rules Management

Business Rule	Where is Implemented
Beds are identified by a ward prefix and a number, for example S01 for Surgery, bed 1.	For this project is fixed in the database. In a future update the system will have the functionality to add beds and the rule will be manage in the application
Bed types include private, semi-private and ward. Ward is usually chosen.	For this project bed types are fixed in the database. In a future update the system will have the functionality to add beds and the rule will be manage in the application The application selects the bed type Ward as default in the admission form.
NLH cannot admit more patients than the total number of available beds in all wards.	Application only add new admission if there is a bed available.
Patients are never deleted.	Application don't give the delete option. Database trigger stops any user from deleting a patient.
When the doctor discharges a patient, the bed to which the patient was assigned is made available.	Application updates the admissions and bed tables when a patient is discharged.
Admission records are never deleted.	Application don't give the delete option. Database trigger stops any user from deleting an admission.
If there is no ward-type bed available in any ward, admissions can assign the patient to a semi-private bed in the preferred ward, if available, at no additional cost. If all semi-private beds are occupied, admissions can assign the patient to a private bed in the preferred ward, if available, at no additional cost.	Application searches for an available ward-type bed and if there is none, automatically upgrades the bed type to semi-private and searches for a semi-private-type bed and if there are none, upgrades again. If there are no beds available in the selected ward, the user must select a different one and the

	application starts the search again.
The patient will pay for a semiprivate or private room if a ward-type bed is available in any ward, but the patient elects a semi-private or private room instead.	If the user selects a semi-private or private bed type, the system will charge as an extra
Surgical patients are automatically admitted to Surgery ward if a bed of the desired type is available. If a bed is not available, the user can select another bed type, or another ward, if available.	The user selects the ward during the admission, if there is not a bed available in the ward the user can select another ward.
Patients 16 years of age and under, if they are not scheduled for surgery, are automatically admitted to Pediatrics ward if a bed of the desired type is available. If a bed is not available, the user can select another bed type, or another ward, if available.	When a patient under 16 years is selected the application selects by default the Pediatrics ward, but the user can change if desired.
Daily rates: Semi-private $267.00, private $571.00, TV $42.50, Phone $7.50.	For this project the rates are fixed in the database. In a future update the system will have the functionality to update extras rates and the rule will be manage in the application
Nurses require a patients list by ward. The report should include the patient ID, name and, at a minimum, the doctor ID. The nurse should be able to select one or all Wards for the report (Use Crystal Reports).	The application has a report that shows those information's about the hospitalized patients. During the development was noted that there is no need for nurses have access to patient's information's that are no longer hospitalized.
The billing report is generated by the administrator. The report should include the admission ID, patient ID, name, and number of days in the hospital. It should also include and identify any additional extra charges. This report requires a total	The application has a form that retrieves those information's when a Admission is selected.

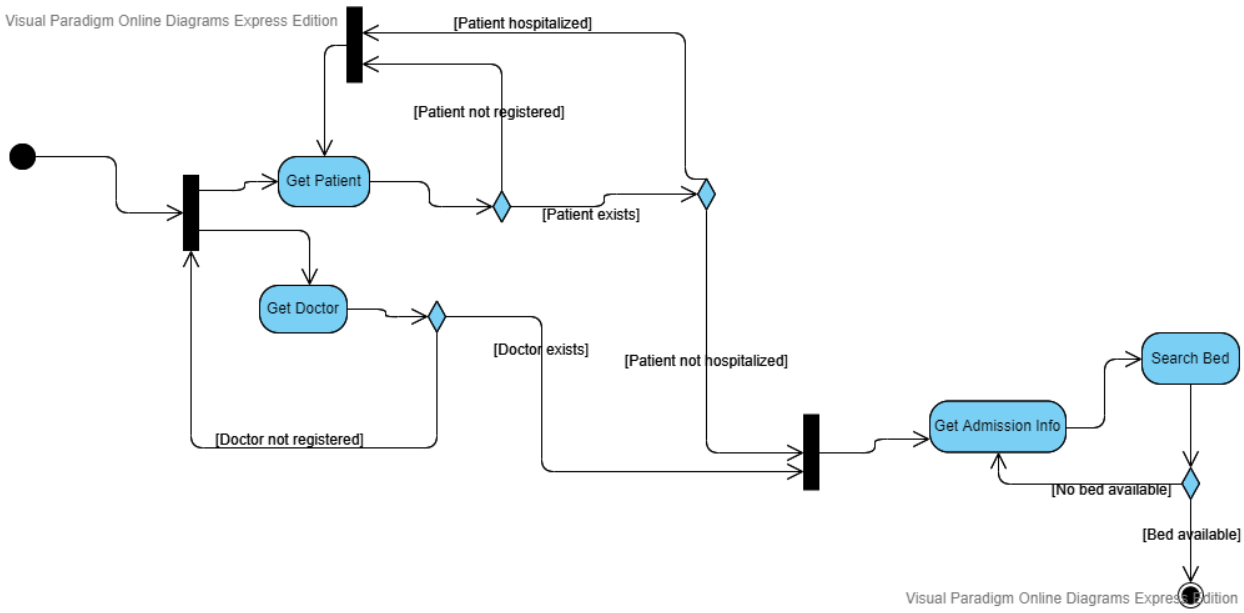
System Design and Programming Logic Design

After applying the database changes, the existing c# code was updated to use the new fields and tables.

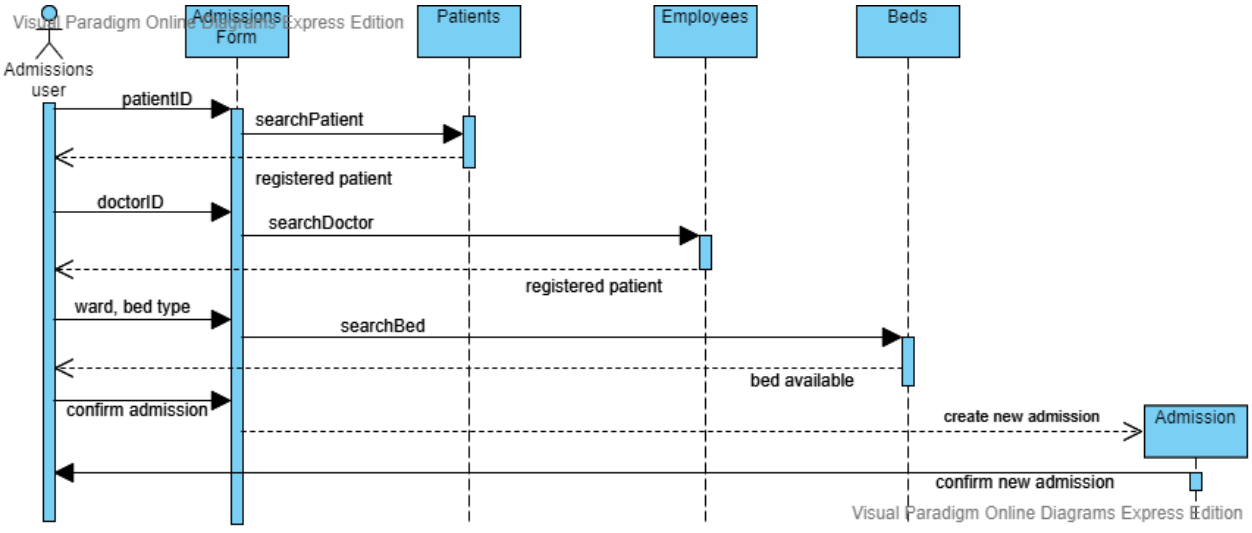
The admission and the reports were developed from scratch but keeping the design pattern of the existent forms.

To help this process the following 3 diagrams were developed as demanded in the project assignment. Since there were no previous material provided to support the creation of the diagram, all three were constructed based on information found on the internet[1].

Activity Diagram

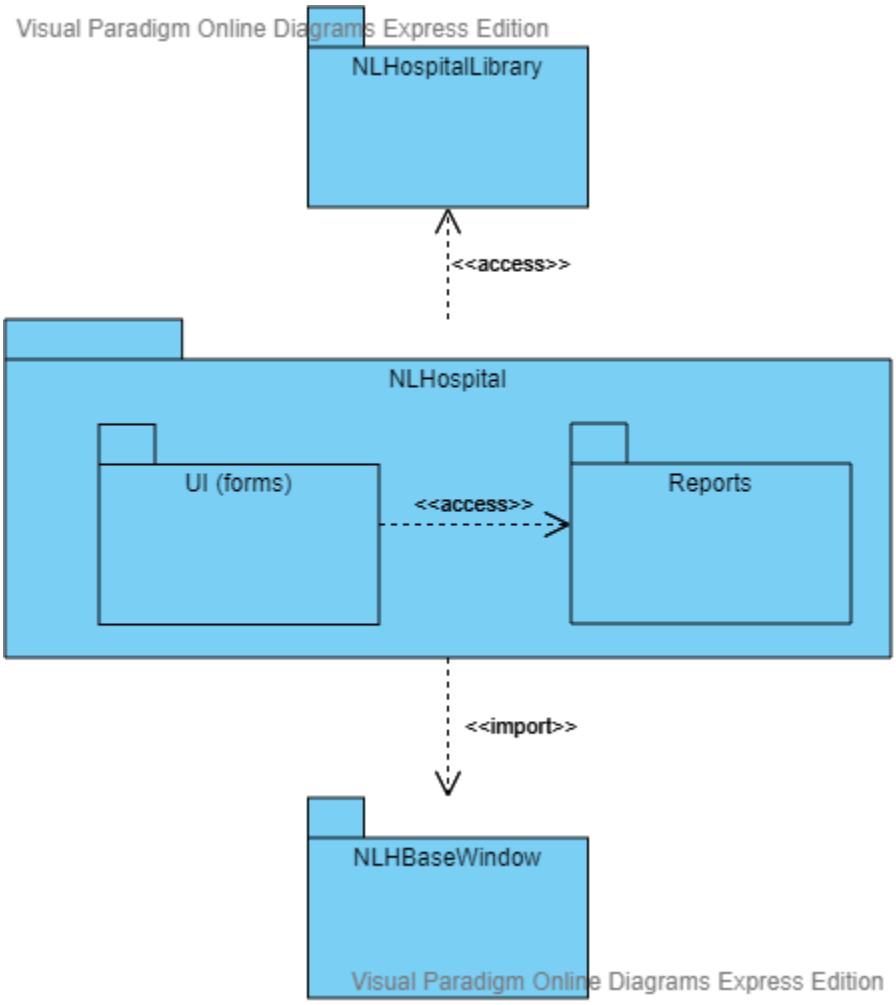


Sequence Diagram



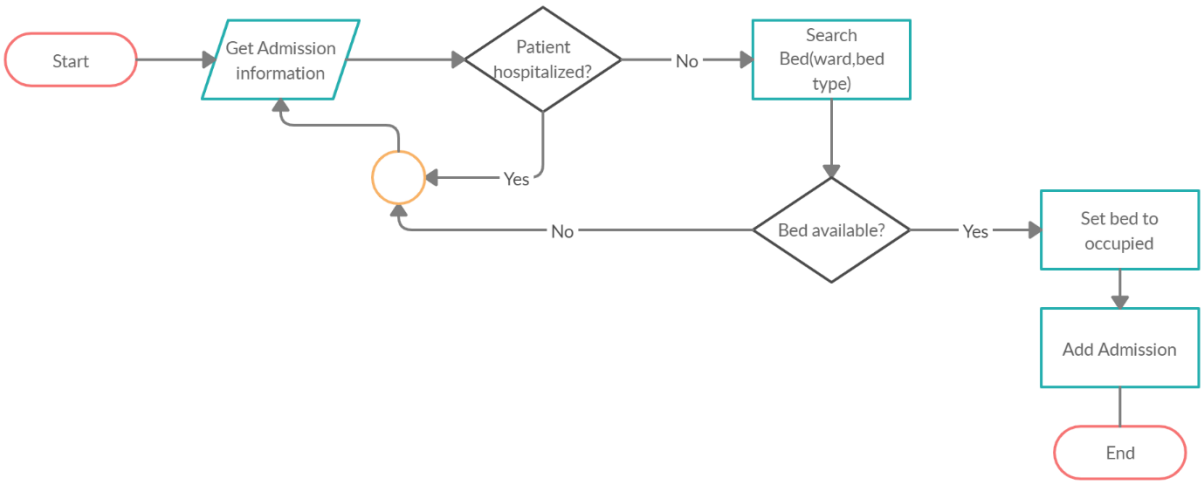
[1] Reference: https://www.visual-paradigm.com/tutorials/

Package Diagram



Admit Patient logic

The following flowchart shows the logic to admit a new patient in the NL Hospital, and the logic to search an available bed it's demonstrated bellow as pseudocodes.



Search bed (ward, bed type)

```
START
bed = FindAvailableBed(ward,bedType)
if (bed = "none available")
{
	msg = "No " + bedType + " bed available in " + ward + ".\n";
	if (bedType = "Ward")
	{
		bed = UpdateBedForFree(ward)
		if (bed != "none available")
		{
			msg += "Patient upgraded bed type at no cost.\n";
			Show user: msg + "Bed " + bed + " found"
			Return bed;
		}
		msg = "No beds available in " + ward + ".\n";
	}
	Show user: msg;
}
else
{
	Show user: "Bed " + bed + " found"
	Return bed
}
END
```

FindAvailableBed(ward,bedType)

```
START
wardChar = ward first letter
bednumber = "none available";
foreach (tempbed in beds table)
{
	if (tempbed ID Starts Withw ardChar)
		if (tempbed type == bedType)
			if (tempbed is not occupied)
				return tempbed ID;
}
return bednumber;
END
```

UpdateBedForFree(ward)

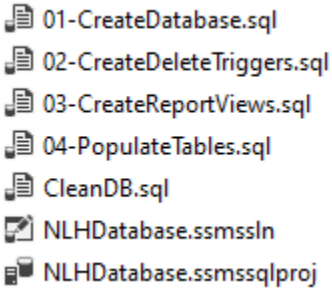
```
START
bed="none available";
bed = FindAvailableBed(ward, "Semi-private");
if (bed == "none available")
       bed = FindAvailableBed(ward, "Private");
return bed;
END
```

Delivery

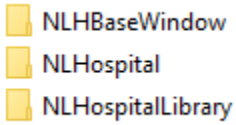
What is included

This project delivery contains beside this document the following:

- A SQL project that has the necessary scripts to implement the database used in the user application (Folder NLHDatabase).



- A visual studio solution containing 3 projects (Folder NLH):
    - A C# library project with the coding to perform the key functions of the NL Hospital called “NLHospitalLibrary”.
    - A project used define the template form for the application called “NLHBaseWindow”. This project was part of the student files provided and it was not changed.
    - The main project that uses both projects above and implements the user’s application for the NL Hospital called “NLHospital”.



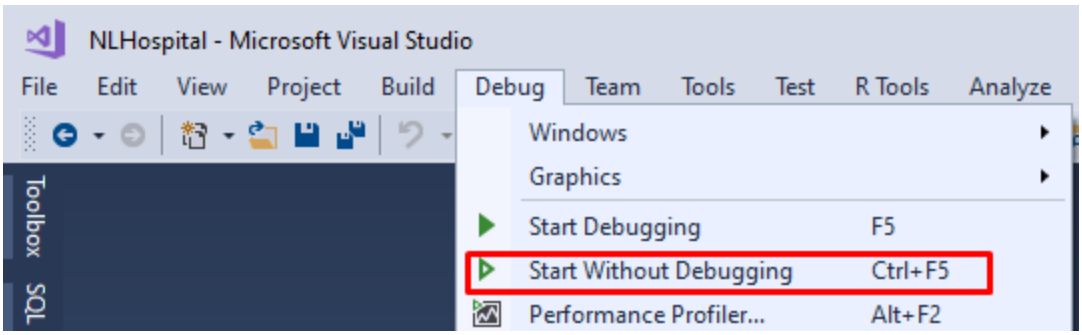
How to execute

Step 1

Run the SQL project scripts following the numeration to create the database.

Step 2

Open the “NLHospital” project and run using the Debug menu.


NLHospital - Microsoft Visual Studio
File
Edit
View
Project
Build
Debug
Team
Tools
Test
R Tools
Analyze
Windows
Graphics
Start Debugging F5
Start Without Debugging Ctrl+F5
Performance Profiler... Alt+F2
Toolbox
SQL