

# **DEVELOPMENT OF AN ONLINE PATIENT SERVICE**

**BY**

**Rafiqul Islam**

**ID: 143-15-4362**

This Report Presented in Partial Fulfillment of the Requirements for the Degree of  
Bachelor of Science in Computer Science and Engineering

Supervised By

**Md. Tarek Habib**

Assistant Professor

Department of CSE

Daffodil International University

Co-Supervised By

**Md. Sadekur Rahman**

Assistant Professor

Department of CSE

Daffodil International University



**DAFFODIL INTERNATIONAL UNIVERSITY**

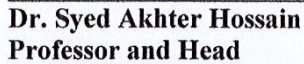
**DHAKA, BANGLADESH**

**DECEMBER 2019**

## **APPROVAL**

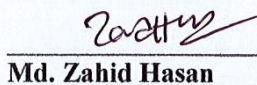
This Project/internship titled “Online Instant Patient Service” submitted by Rafiqul Islam, ID NO: 143-15-4362 to the Department of Computer Science and Engineering, Daffodil International University has been accepted as satisfactory for the partial fulfillment of the requirements for the degree of B.Sc. in Computer Science and Engineering and approved as to its style and contents. The presentation has been held on 07 December 2019.

## **BOARD OF EXAMINERS**

  
\_\_\_\_\_

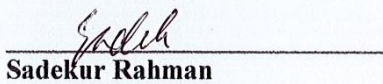
**Dr. Syed Akhter Hossain**  
**Professor and Head**  
Department of Computer Science and Engineering  
Faculty of Science & Information Technology  
Daffodil International University

**Chairman**

  
\_\_\_\_\_

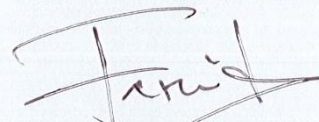
**Md. Zahid Hasan**  
**Assistant Professor**  
Department of Computer Science and Engineering  
Faculty of Science & Information Technology  
Daffodil International University

**Internal Examiner**

  
\_\_\_\_\_

**Sadekur Rahman**  
**Assistant Professor**  
Department of Computer Science and Engineering  
Faculty of Science & Information Technology  
Daffodil International University

**Internal Examiner**

  
\_\_\_\_\_

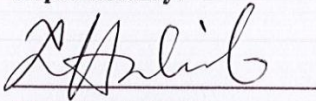
**Dr. Dewan Md. Farid**  
**Associate Professor**  
Department of Computer Science and Engineering  
United International University

**External Examiner**

## DECLARATION

I hereby declare that, this Internship has been done by me under the supervision of **Md. Tarek Habib, Assistant Professor and, Department of CSE**, and Daffodil International University. I also declare that neither this project nor any part of this report has been submitted elsewhere for award of any degree or diploma.

**Supervised by:**



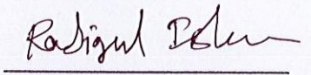
**Md. Tarek Habib**  
Assistant Professor  
Department of CSE  
Daffodil International University

**Co- Supervised by:**



**Md. Sadekur Rahman**  
Assistant Professor  
Department of CSE  
Daffodil International University

**Submitted by:**



**Rafiqul Islam**  
ID: 143-15-4362  
Department of CSE  
Daffodil International University

## ACKNOWLEDGEMENT

First, we express our heartiest thanks and gratefulness to almighty God for His divine blessing makes us possible to complete the final year project/internship successfully.

We really grateful and wish our profound our indebtedness to **Supervisor Md. Tarek Habib, Assistant Professor**, Department of CSE Daffodil International University, Dhaka. Deep Knowledge & keen interest of our supervisor in the field of “**Software Engineering**” to carry out this project. His endless patience, scholarly guidance, continual encouragement, constant and energetic supervision, constructive criticism, valuable advice, reading many inferior draft and correcting them at all stage have made it possible to complete this project.

We would like to express our heartiest gratitude to the Almighty Allah and **Prof. Dr. Syed Akhter Hossain, Head**, Department of CSE, for his kind help to finish our project and also to other faculty member and the staff of CSE department of Daffodil International University.

We would like to thank our entire course mate in Daffodil International University, who took part in this discuss while completing the course work.

Finally, we must acknowledge with due respect the constant support and patience of our parents.

## ABSTRACT

Prompt patient service is very important for patient's treatment and sometimes lifesaving issues for a critical patient. In a hospital especially in the third world countries patients are suffering a lot for getting prompt medical facilities. Sometimes the admitted patient can't call duty Doctors or Nurses in emergency purpose. On the other hand, duty doctor or nurses are also not identified critical patients instantly. This thing makes serious hamper of good patient service. We offer an **online instant patient service**, where a hospital admitted patient can call duty doctor and duty nurses at any time from Mobile, Tab or any Smart Display provided by Hospital authority or patient's personal device. Doctors and Nurses also getting the call information instantly and visit the patient's room and take necessary steps faster. Through our system, Doctor can write prescription, diet instruction diagnosis report etc. Nurses also can write the Observation Report, bills etc. Patient also can view all the above info instantly. In this way our system can ensure an excellent service to the patient with a very nominal effort.

# TABLE OF CONTENTS

<b>CONTENTS</b>	<b>PAGE</b>
Board of Examiners	i
Declaration	ii
Acknowledgement	iii
Abstract	iv
List of Tables	vi
List of Figures	vii
<b>CHAPTER 1: Introduction</b>	<b>1-2</b>
1.1 Introduction	1
1.2 Motivation	1
1.3 Objectives	1
1.4 Expected Outcome	1
1.5 Report Layout	2
<b>CHAPTER 2: Background</b>	<b>3-7</b>
2.1 Introduction	3
2.2 Related Works	3
2.3 Comparative Studies	3
2.4 Scope of the Problem	3
2.5 Challenges	7
<b>CHAPTER 3: Requirement Specification</b>	<b>8-10</b>
3.1 Business Process Modeling	8
3.2 Requirement Collection and Analysis	8
3.3 Use Case Modeling and Description	8
3.4 Logical Data Model	9

3.5 Design Requirements	10
<b>CHAPTER 4: Design Specification</b>	<b>14-18</b>
4.1 Overview of Design and Development	14
4.2 Front-end Design	15
4.3 Back-end Design	16
4.4 Interaction Design and UX	17
4.5 Implementation Requirements	18
<b>CHAPTER 5: Implementation and Testing</b>	<b>17-25</b>
5.1 Implementation of Database	19
5.2 Implementation of Front-end Design	21
5.3 Implementation of Interactions	24
5.4 Testing Implementation	24
5.5 Test Results and Reports	25
<b>CHAPTER 6: Conclusion and Future Scope</b>	<b>26-30</b>
6.1 Discussion and Conclusion	26
6.2 Scope for Further Developments	26
References	27
Appendices	32
Appendix A: Project Reflection	

## LIST OF FIGURES

<b>FIGURE</b>	<b>PAGE</b>
Figure 3.3.1: Use Case Diagram	9
Figure 3.4.1: Logical Mode	10
Figure 3.5.1: DFD	13
Figure 4.1.1: Water fall Model	14
Figure 4.3.1: Table Structure	16
Figure 4.1: Total Database ERD	17
Figure 5.1.1: Patient Info	19
Figure 5.1.2: Call Doctor	19
Figure 5.1.3: Call Nurse	20
Figure 5.1.4: Diagnosis	20
Figure 5.1.5: Prescription	20
Figure 5.2.1: Homepage	21
Figure 5.2.2: user info/sign	21
Figure :5.4.1 Doctor Portal	22
Figure 5.5.1: Nurse Portal	23
Figure 5.6.1: Patient Portal	23



# **CHAPTER 1**

## **INTRODUCTION**

### **1.1 Introduction**

Admitted Patients into a hospital faces different problems like call for a Doctor or Nurses in case of emergency. This thing is happening when there is no attendant for a patient. Sometimes, the patients or the guardians are not getting the prescription, nurse observation report, diagnosis report etc. instantly. We try to develop a system which can provide above service through online. Patient, Doctor and Nurse are connected with our system and write their update over there and need not to maintain any hard copy or do any paper work.

### **1.2 Motivation**

At present time there are some companies who are providing hospital management system or treatment management system but no one consider the Admitted Patient Service like us. Our concentration is to provide only the Hospital Admitted Patient.

### **1.3 Objectives**

We proposed an application in where Patient, Doctor, Nurse can get related info from different part of our application. But if he/she wants to put some info in any part, he/she must need to confirm his/her Sign in and create an account in our system. The Sign in process is very easy and instantaneous job. After successful Sign in, any person can put any info in related area. An authorized person can change his/her info at any time and the other user can get the reflection of the change data accordingly.

### **1.4 Expected Outcome**

After login into our application, a related user can put related part like call for a Doctor, call for a Nurse, Create Prescription, Diagnostic Report, Nurse Observation, and Diet Plan etc. After that, any visitor can get all above info at any time and communicate with related user. After successful dealing, a user can inactive his info or make any change of his/her info at any time also.

## **1.5 Report Layout**

At first, when an user opens our site he/she will get a home page consisting a login feature which is only for admin. Admin can log in using username and password. By inputting valid user id and password and pressing login button. On the other hand if the user is a patient, he/she need not to login into our system and can get all medicine finding options automatically. After finding/ selecting any medicine, we can apply for order and finally our system confirm the order by sending a confirmation email to the patient's email inbox.

## **CHAPTER 2**

### **Background**

#### **2.1 Introduction:**

Online Instance Patient Service is an automated web-based application system. There is no need to install the application. To get any information from this application a user needs to smart device our application but for put any information a user needs to log in into our system through any browser.

#### **2.2 Related Works:**

Right now, very few organizations are using smart patient management and medical service in the world but most of them are desktop based application and can be access within the hospital campus. But we offer Online Patent management System which can be accessed from anywhere through internet.[9], [10]

#### **2.3 Comparative Studies:**

Accomplishment challenges are this type of task which is really difficult to overcome. But things can be solved. There are a lot of challenges came ahead to develop the Online Instant Patient Service such as security issues, personal information sharing, fake data input etc.

#### **2.4 Scope of the Problem:**

Using Online Instant Patient Service, a patient can get following facilities:

Patient can:

- Call for a doctor
- Call for Nurse
- View Prescription
- View Diagnosis Report
- View Diet Chart
- View Bills
- View Patient Observation

### **Call for a doctor:**

The medium of Online Instance Patient Service application is getting very fast services. In-Door and Out-Doors will be able to get all the specialist services. The physician needs to call the doctor in need.

### **Call for a Nurse:**

The nurse will always be ready to serve the patient. Nurse patient will always care for her to provide Patient's services. And when the need for any of the services of the Patient will give the nurse message or call, and the nurse will be engaged in that time.

### **View Prescription:**

High-tech Patient Service is a modern application of fast-growing patients. Patients can take modern services through this application. When the doctor prescription a patient, then prescription through high-tech patient service if you want to see the patient's prescription.

### **View Diagnoses Reports:**

The doctor will see the patient and the doctor prescribes the diagnosis test for the diagnosis of the patient's disease. And the Patient Diagnosis test. The patient will not have to wait long time for the diagnosis report. If you want to see the Patient Diagnosis report, you can easily see through High-Tech Patient Services.

### **View Diet Chart:**

Health is the root of all happiness, so we need to routinely measure the routine to maintain health. And the doctor advises a patient with a healthy diet routine. And patients need to run routine routines. If you want to see the doctor's diet chart hi-tech.

### **View Bills:**

The present era is the age of technology. People of this era are stagnant on the Internet. So, in a short time, the information about the patient's bill, the date of the council, and any other information from this place will be known through this application.

### **Using Online Instance Patient Service, a doctor can get following facilities:**

#### **Doctor can do:**

- Prescription.
- Diet Chart
- View Call List

- Call for a Nurse
- View Diagnoses Report
- View Patient Observation Report

### **Prescription:**

The main objective of our application is to give quick service to the patient and provide all types of information about the patient. In this application, when a patient takes the service, the doctor will consult and the prescription will be done on the condition of the patient and the patient will take the service

### **Diet Chart:**

Everything in human life goes according to a rule. So to maintain our physical condition, we have to eat food by following the rules. The doctor will ban the food that will disturb the physical condition of the food. And according to the doctor's diet chart, Patient's original goal.

### **View Call List:**

Using the Patient Service application, the doctor will be able to know about the condition of the patient from his room. If a patient or nurse wishes to contact the doctor if necessary, you can send a message or call to the view call list. And the doctor can see the call list from his chamber.

### **Call for a Nurse:**

We know that the main goal of the nurse is to cure the patient faster by serving the patient. And the nurse does this job. The doctor will always half the nurse. If a doctor requires a help, the doctor will call the nurse to the nurses.

### **View Patient Observation Report:**

Generally, we know, 24 hours of supervision is done by a doctor and a nurse patient to improve the patient's health. And the nurse will review the 24-hour patient and report it. The doctor will be able to see the depression report through this application about the condition of the patient.

## **Using Online Instance Patient Service, a Nurse can get following facilities:**

### **Nurse can:**

- View Call List
- Patient Observation Report
- Diagnoses Report Entry
- Bill Entry
- Call for a Doctor

### **View Call List:**

The goal of our application is to send a message or information briefly. Patient's physical condition may improve or deteriorate. When the patient's service or something else will be needed. Then the Patient Nurse can give a message or call. And the nurse will be able to see the call list through our application.

### **Patient Observation Report:**

The nurse will always keep the patient in the observation and report the progress of the patient's physical condition. According to the 24-hour observation list of the patient, the patient will be provided with the services. The Patient Observation List nurse can see from his room.

### **Diagnoses Report Entry:**

For diagnosing the disease, the diagnosis test will be done by practicing the condition of the patient. Patient will test the diagnosis. After completing the diagnosis report, the nurses will enter the report. The doctor and nurse can see an attached report.

### **Bill Entry:**

Nurse can enter the bill through our application. Medical, O, T charge, Emergency Diagnosis Test, Injection etc. The bills will enroll the nurses. Through the online application, you can see the Patient Bill.

### **Call for a Doctor:**

Nurses will always keep the patients at the observation. Patient's physician will tell the doctor about the physical progress, medicine chart, and diet chart doctor. Nurses can call the doctor if the patient's physical condition is bad.

## **2.5 Challenges:**

The main challenges of our proposed service is to confirm it 24 hours through online. Security is another concern about any online service. But we try our best to confirm those.

## **CHAPTER 3**

### **Requirement Specification**

#### **3.1 Business Process Modeling:**

Our offered platform is followed totally dynamic web-based rule. In this system, we used Web GUI (Graphical User Interface) technology for client-side web programming, C# for server-side programming, and SQL Server 2008 for Database. If anyone interested in gaining practical and comprehensive insight into the C# programming language and SQL Server database and this prominent technology can be used together to create dynamic, database driven web application, this project can be the great help for them. [1], [2]

#### **3.2 Requirement Collection and Analysis**

We try to accumulate all the requirements in a format so that we can identify the core & non-core requirements and our technical people can draw a design for the software.

##### **Core Requirements:**

- Patient can Call Doctor or Nurses for any problem
- Patient can get the Prescription, Diagnosis Report, Diet Chart and Bills
- Patient can put there Feedback

##### **Non-Core Requirements:**

- Doctor can Prescribe Medicine, Diagnosis and Diet
- Nurses can Entry Observation Report and view Prescribe Diet & Medicine

#### **3.3 Use Case Modeling and Description**

A use case diagram is a dynamic or behavior diagram in UML. Use cases are a set of Actions, Services, and functions that the system needs to perform. Identifying Actors and their Activity with graphical representation is the main goal of Use Case Design.

In our System there are two Actors: Patient, Doctor, Nurses.



And there are 7 Activity: Call for a doctor, Call for Nurse, View Prescription, View Diagnosis Report, View Diet Chart, View Bills, and View Patient Observation. The following figure: 3.3.1 has been shown use case diagram.

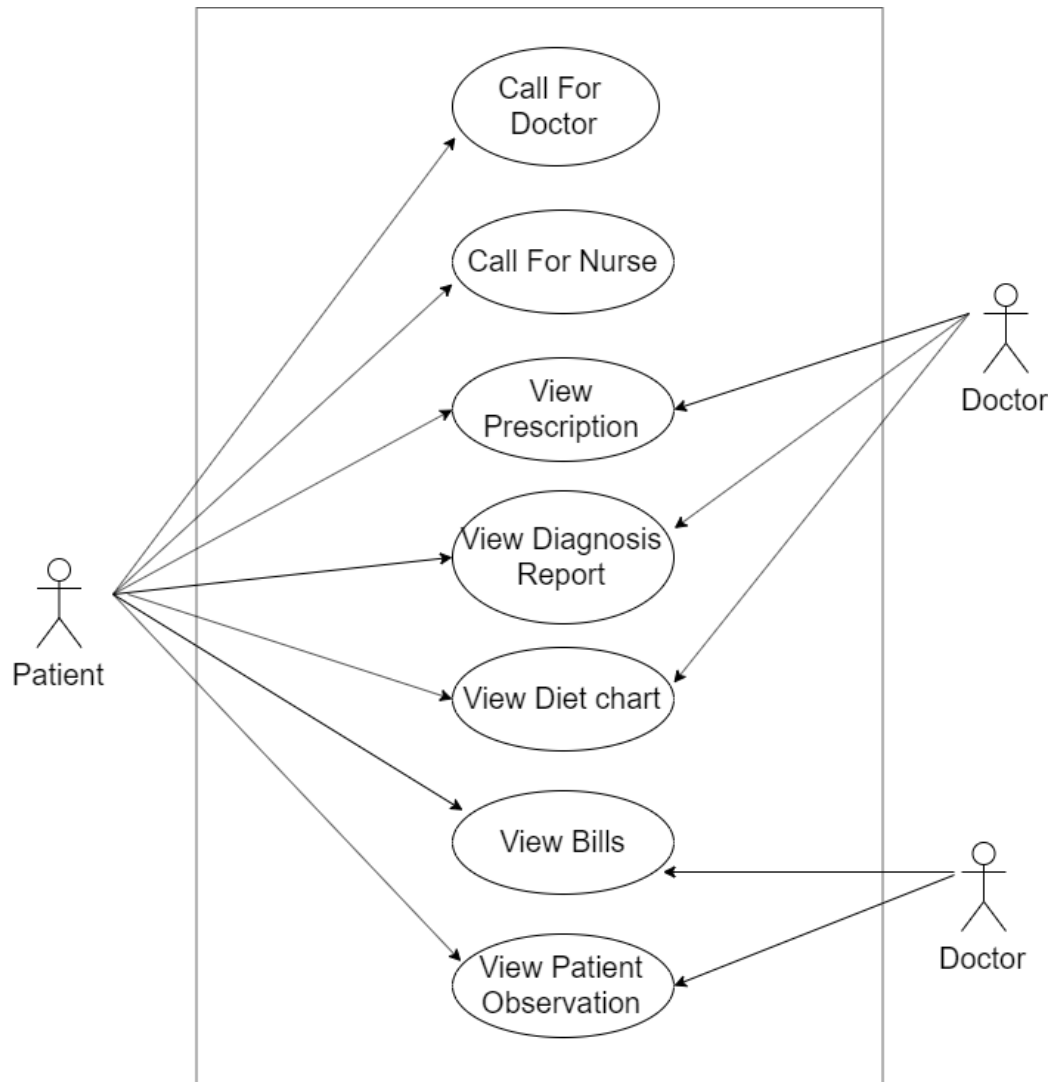


figure: 3.3.1 use case diagram

### 3.4 Logical Data Model:

All the Actors are connected with their activity. For example a patient can call doctor or nurses but he/she can view prescription only when a doctor prescribe anything on the system. Similarly a nurse can view the prescription only when the doctor write something on the system.

The following fig: 3.4.1: has been shown Logical Mode

.

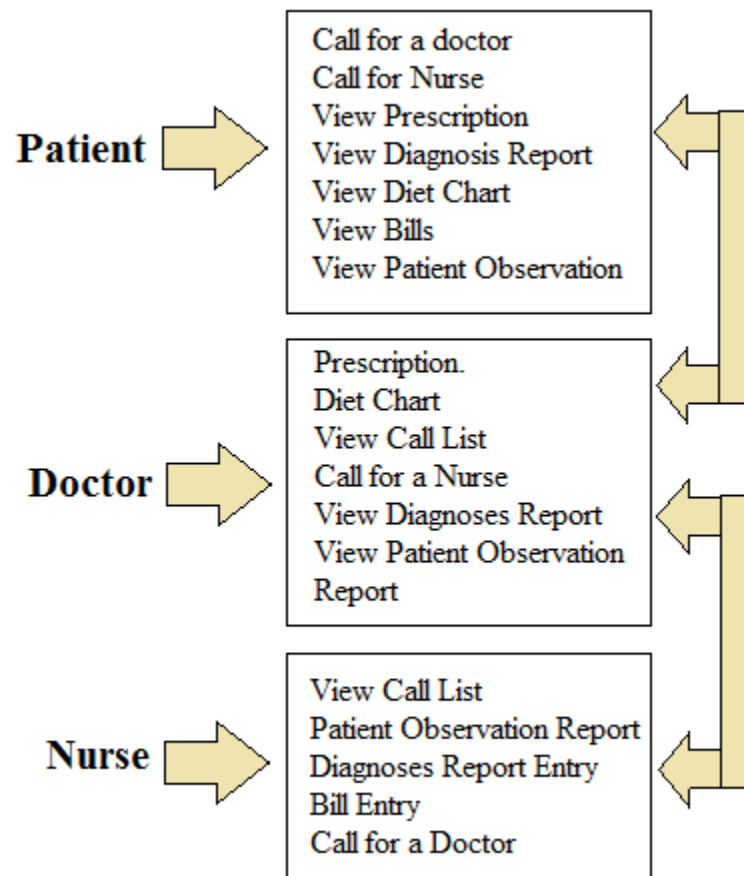
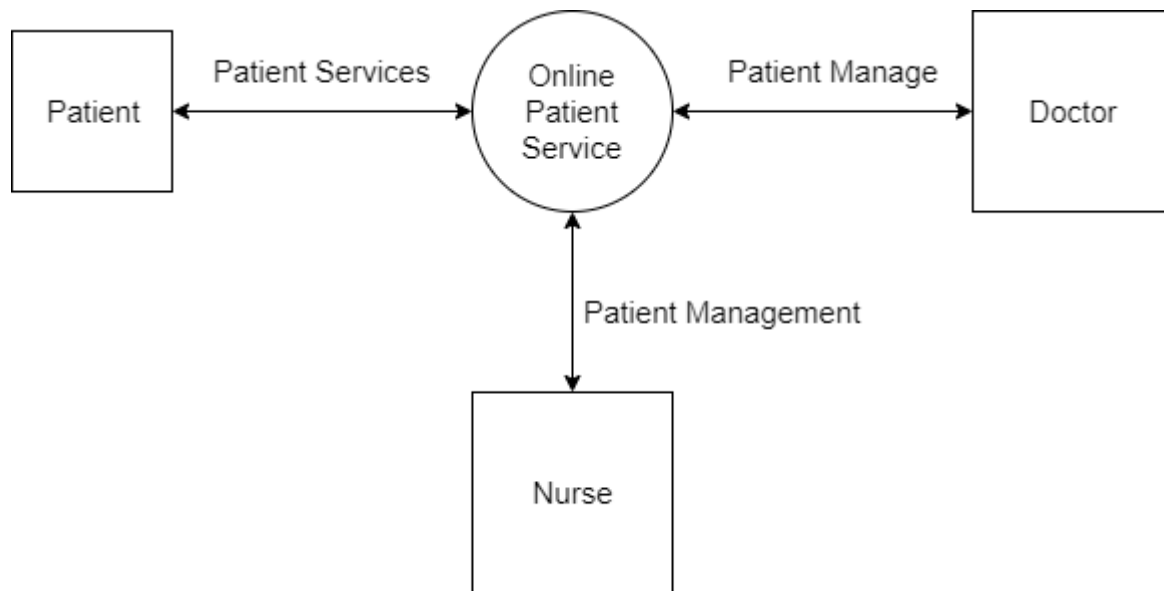


Figure 3.4.1: Logical Mode

### 3.5 Design Requirements:

We need to design a Relational Database for our projects and for confirm web based online application. User Interface will be interactive and User Friendly. Thus we use Web-GUI Framework which, which confirm a very user friendly UI. Web-GUI gives Forms give Desktop flavor but actually works as a web application.



The following Figure 3.5.1: has been shown zero level data flow diagram (DFD) .

Figure 3.5.1: Zero Level(DFD)

The following Figure 3.5.2: has been shown one level data flow diagram (DFD) .

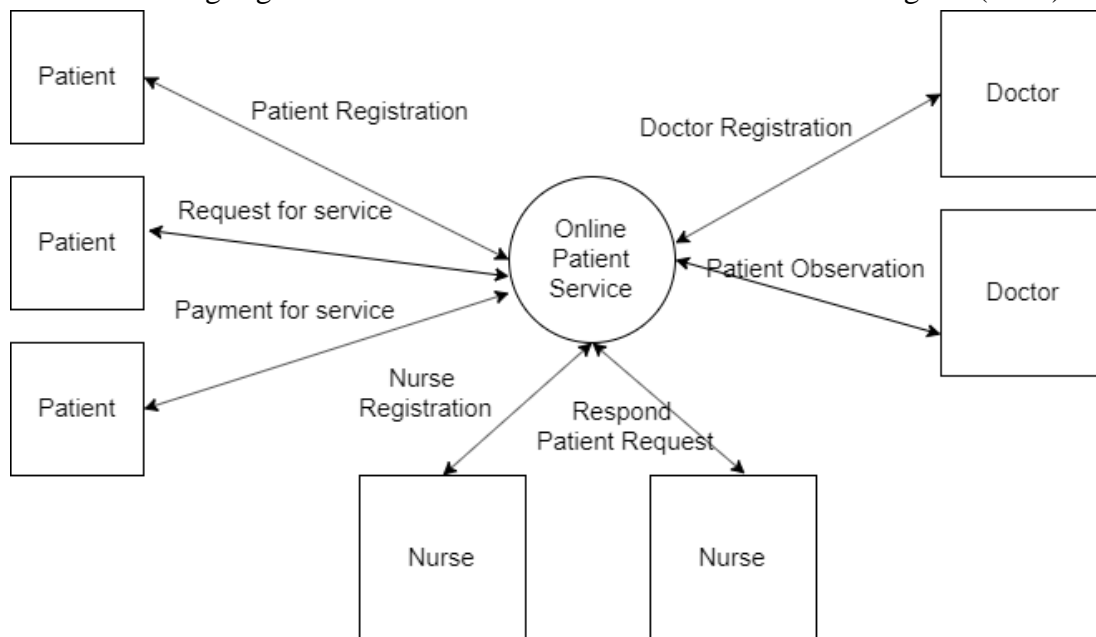


Figure 3.5.2: DFD (one Level)

The following Figure 3.5.3: has been shown 2nd level data flow diagram (DFD).

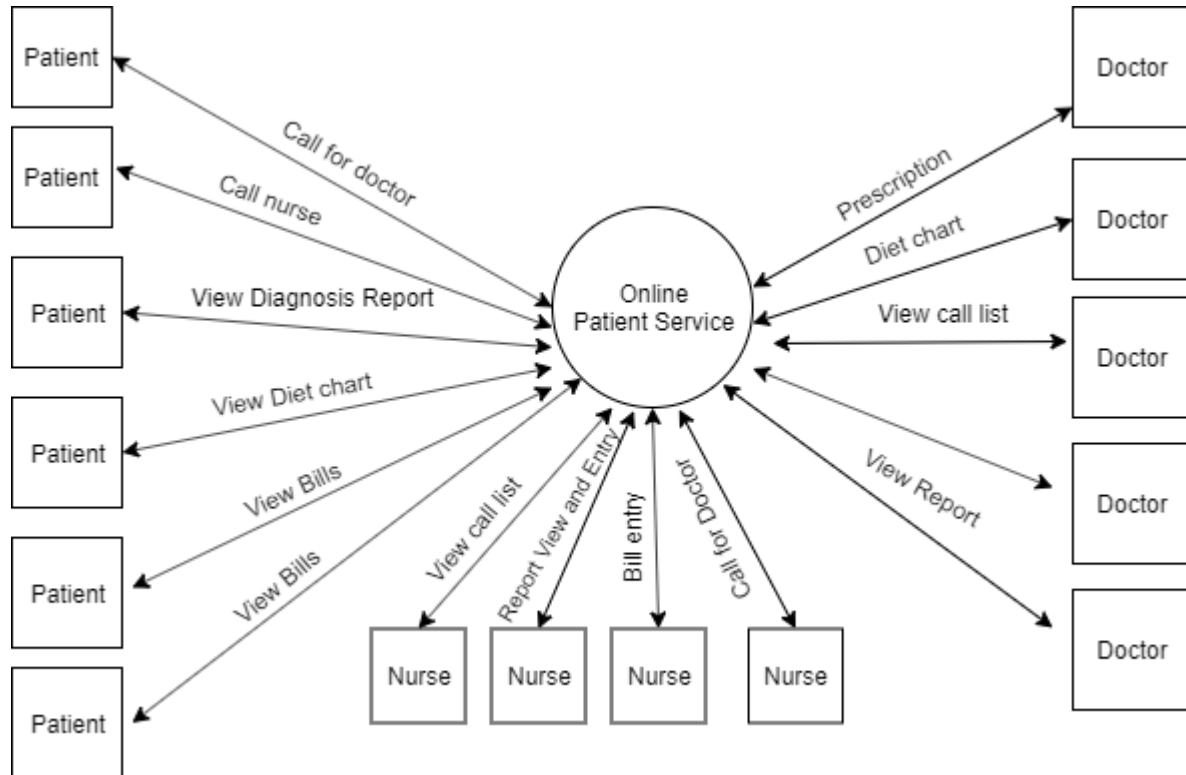


Figure 3.5.3: Second Level (DFD).

The following Figure 3.5.4: has been shown data flow diagram (DFD) .

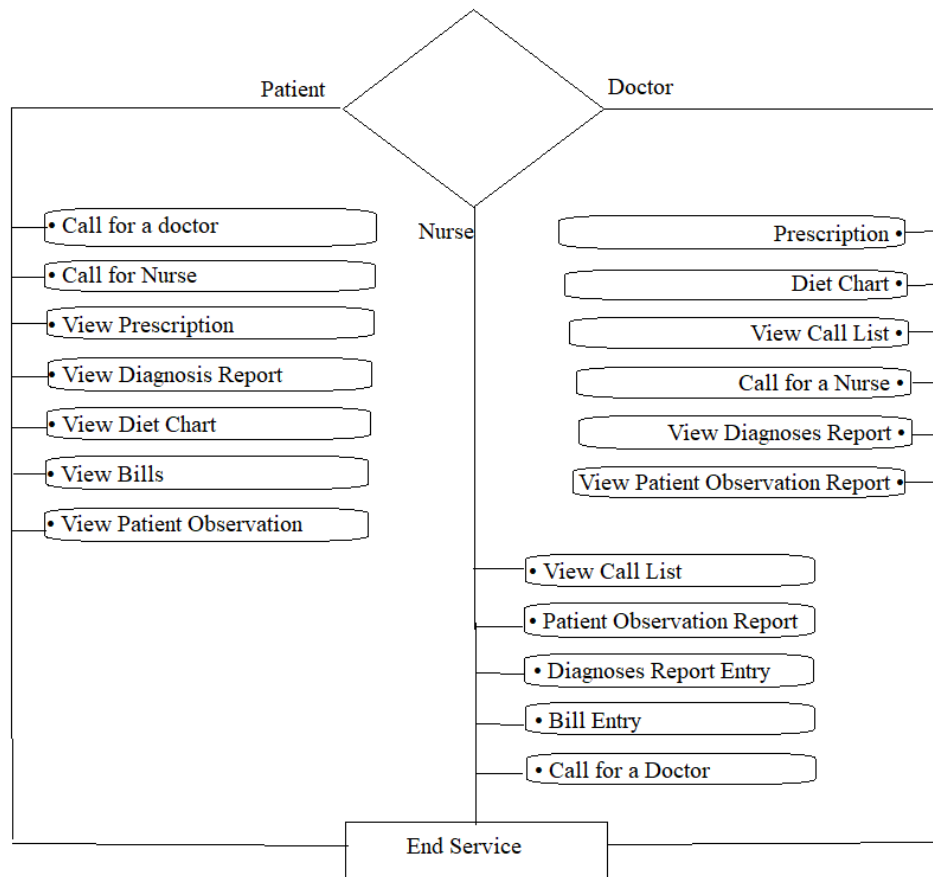


Figure 3.5.4: data flow diagram

## CHAPTER 4

### Design Specification

#### 4.1 Overview of Design and Development

We develop Automated Question Generator using Microsoft Platform. We use Microsoft .Net web Application, C# .Net and Microsoft SQL Server 2005. First of all we try to collect main requirement by consulting among us and some other students. Then try to create the Interfaces. Based on the requirements and Templates of Interfaces we create Database. Finally we Design the UI and related methods and functions [3], [4].

After completing the development part, we do Testing and finally close the development part and keep it ready for use. In our Project, we use Waterfall Model. The following fig: 4.1.1: has been shown water fall model [5].

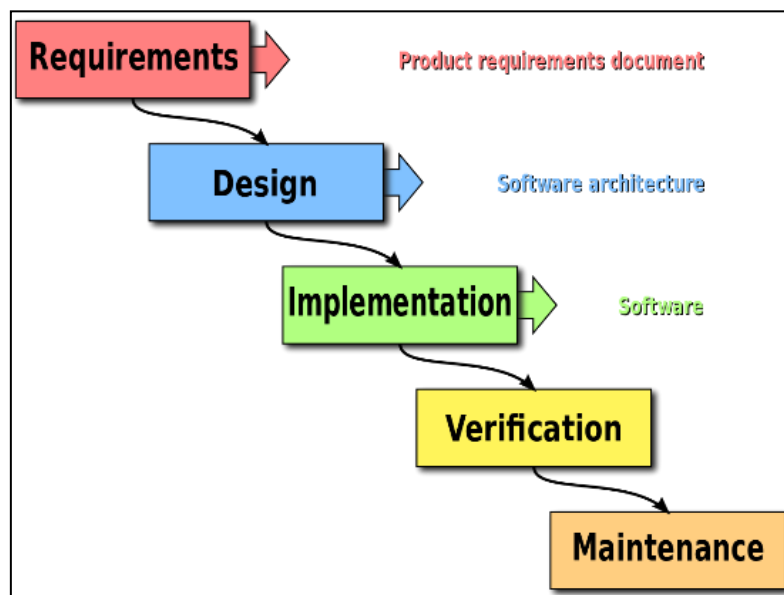


Figure 4.1.1: Water fall Model

## 4.2 Front-end Design:


# Online Instance Patient Service

Create New User

Login

User ID:

Password:




User Info

Full Name:  Phone:

Address:  Email:

Expected User Name:  User Type:

Login Password:  Password Again:



Activate Windows  
Go to PC settings to activate Windows.

Hello Adam

Welcome to Doctor Portal

 Prescription

 View Patient Call List

 Nurse Observation Report


 Payment/ Bill


 Diagnosis Report


 Diet Plan


Hello Khatarin


Welcome to Nurse Portal


 Prescription

 View Patient Call List

 Nurse Observation Report


 Payment/ Bill


 Diagnosis Report

 Diet Plan

Hello Jamal

Welcome to Patient Portal


 Prescription


 Payment/ Bill

 Diagnosis Report

 Diet Plan

 Nurse Observation Report

 Call Duty Doctor

 Call Duty Nurse

## 4.3 Back-end Design

In this system, we used database named “Hospital” which consists four following tables: The following fig: 4.3.1 has been Table Structure.

UserInfo *			
	Column Name	Data Type	Allow Nulls
🔑	UserID	int	<input type="checkbox"/>
	UserName	nvarchar(50)	<input checked="" type="checkbox"/>
	Type	nvarchar(50)	<input checked="" type="checkbox"/>
	Address	nvarchar(50)	<input checked="" type="checkbox"/>
	Phone	nvarchar(50)	<input checked="" type="checkbox"/>
	Email	nvarchar(50)	<input checked="" type="checkbox"/>
	ExpectedUserID	nvarchar(50)	<input checked="" type="checkbox"/>
	Password	nvarchar(50)	<input checked="" type="checkbox"/>
			<input type="checkbox"/>

DiagnosisInfo *			
	Column Name	Data Type	Allow Nulls
🔑	DiagnosisID	int	<input type="checkbox"/>
	PatientID	int	<input checked="" type="checkbox"/>
	DiagnosisReport	nvarchar(150)	<input checked="" type="checkbox"/>
	Time	datetime	<input checked="" type="checkbox"/>
	NurseID	int	<input checked="" type="checkbox"/>
	PatientName	nvarchar(150)	<input checked="" type="checkbox"/>
			<input type="checkbox"/>

Prescription *			
	Column Name	Data Type	Allow Nulls
🔑	PrescriptionID	int	<input type="checkbox"/>
	PatientID	int	<input checked="" type="checkbox"/>
	DoctorID	int	<input checked="" type="checkbox"/>
	Description	nvarchar(150)	<input checked="" type="checkbox"/>
	Time	datetime	<input checked="" type="checkbox"/>
	PatientName	nvarchar(150)	<input checked="" type="checkbox"/>
			<input type="checkbox"/>

DietInfo *			
	Column Name	Data Type	Allow Nulls
🔑	DietID	int	<input type="checkbox"/>
	PatientID	int	<input checked="" type="checkbox"/>
	DietInfo	nvarchar(150)	<input checked="" type="checkbox"/>
	Time	datetime	<input checked="" type="checkbox"/>
	DoctorID	int	<input checked="" type="checkbox"/>
	PatientName	nvarchar(150)	<input checked="" type="checkbox"/>
			<input type="checkbox"/>

CallNurse *			
	Column Name	Data Type	Allow Nulls
🔑	CallNurseID	int	<input type="checkbox"/>
	Problem	nvarchar(150)	<input checked="" type="checkbox"/>
	Time	datetime	<input checked="" type="checkbox"/>
	PatientID	int	<input checked="" type="checkbox"/>
	PatientName	nvarchar(150)	<input checked="" type="checkbox"/>
			<input type="checkbox"/>

DiagnosisInfo *			
	Column Name	Data Type	Allow Nulls
🔑	DiagnosisID	int	<input type="checkbox"/>
	PatientID	int	<input checked="" type="checkbox"/>
	DiagnosisReport	nvarchar(150)	<input checked="" type="checkbox"/>
	Time	datetime	<input checked="" type="checkbox"/>
	NurseID	int	<input checked="" type="checkbox"/>
	PatientName	nvarchar(150)	<input checked="" type="checkbox"/>
			<input type="checkbox"/>

Figure 4.3.1: Table Structure



## 4.4 Interaction Design and UX

The Final result of ER-modeling is called the ER-Diagrams (ERD). ER-modeling is a data modeling technique used in software engineering to produce a conceptual data model of an information system. Diagrams created using this ER-modeling technique are called Entity-Relationship Diagrams, or ER diagrams or ERDs. So you can say that Entity Relationship Diagrams illustrate the logical structure of databases. Dr. Peter Chen is the originator of the Entity-Relationship Model. His original paper about ER-modeling is one of the most cited papers in the computer software field. Currently the ER model serves as the foundation of many system analysis and design methodologies, computer-aided software engineering (CASE) tools, and repository systems. The following Figure 4.1 has been Total Database ERD [6], [7].

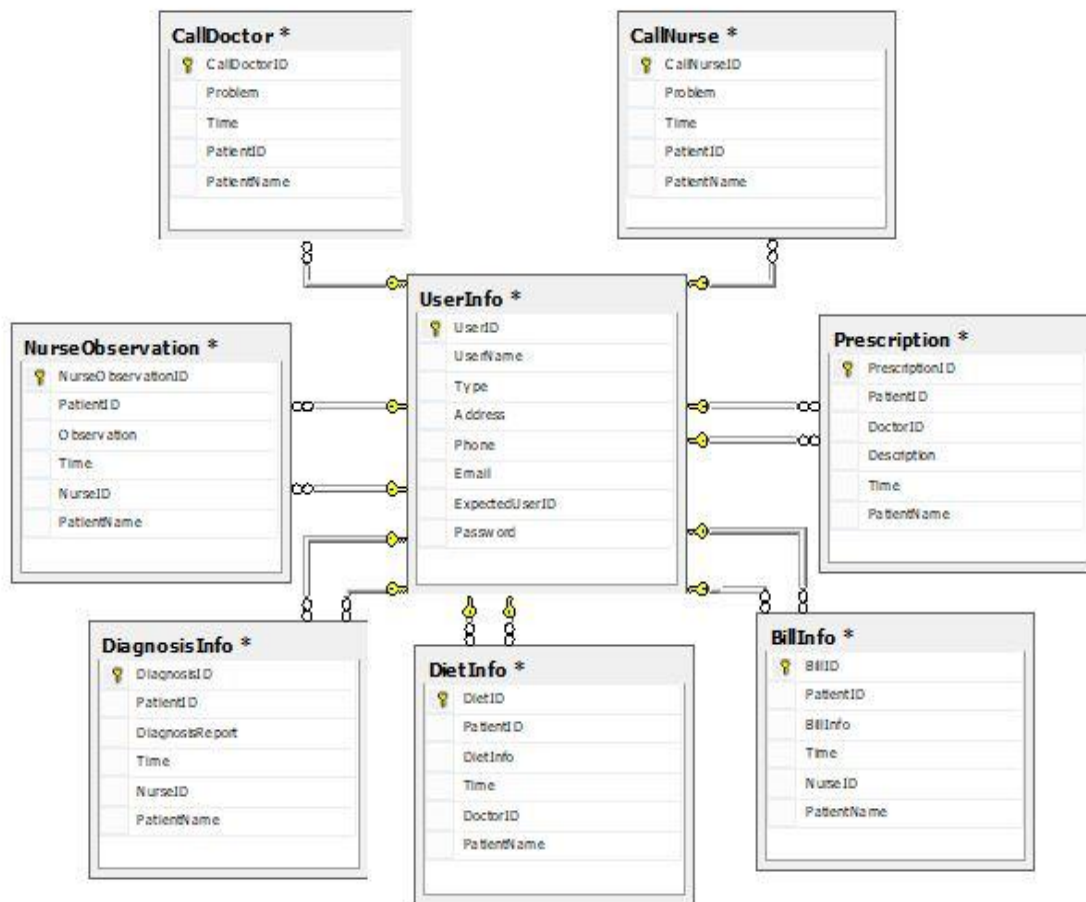


Figure 4.1: Total Database ERD

## 4.5 Implementation Requirements

The different phases of our Development times are:

- Requirement Analysis
- Template Design
- Database Design
- UI Design
- Coding (Method and Function)
- Testing

Database server consisting with a computer program and it provides database services to other computers; mainly depend on the client-server model. The term used to refer to the back-end system of a database application using server architecture. These back-ends sometimes called a database server. Database server performs tasks as data analysis, storage, data-manipulation, archiving, backup and recovery also.

## CHAPTER 5

### Implementation and Testing

#### 5.1 Implementation of Database

We develop Automated Question Generator using Microsoft Platform. We use Microsoft .Net web Application, C# .Net and Microsoft SQL Server 2008. First of all we try to collect main requirement by consulting among us and some other students. Then try to create the Interfaces. Based on the requirements and Templates of Interfaces we create Database. Finally we Design the UI and related methods and functions. The following Figure 5.1.1 has been Patient Info.

PatientName	Age	Mobile	ExpectedDate	Email	Med ID
Md. Kamal	30	01868686	2019-01-01 00:...	abc@gmail.com	1
Md. Abul	45	034324324	2019-01-01 00:...	abf@gmail.com	1
Bikas	64	01868686	2019-01-01 00:...	abc@gmail.com	1
Badol	22	045464564	2019-01-01 00:...	abc@gmail.com	1
Kajol	30	0453454	2019-01-01 00:...	abc@gmail.com	1
Hamid	4	05656	2019-01-01 00:...	abc@gmail.com	1
NULL	NULL	NULL	NULL	NULL	NULL

Figure 5.1.1: Patient Info

A Call Doctor table mainly contain Problems and Time with the Doctor's name.

CallDoctorID	Problem	Time	PatientID	PatientName
1	Pain	2017-12-05 01:00:20.213	7	Abul
2	Fever	2017-12-05 01:04:42.673	7	Malik
3	Headech	2017-12-06 01:03:15.700	7	Jamal
4	Blood Presseure	2017-12-06 10:47:31.663	7	Kamla
5	Kidney	2017-12-06 12:08:38.320	7	Titul
6	Hipertention	2017-12-06 15:45:07.840	7	Jamal
7	I am feeling sick	2017-12-07 11:56:07.957	8	Alam
8	sick	2017-12-07 12:00:29.613	8	Alam

Figure 5.1.2: Call Doctor

A Nurse Call table mainly contain Problems and Time with the Nurse name.

CallNurseID	Problem	Time	PatientID	PatientName
1	Problem	2017-12-05 01:35:42.360	7	Jamal
2	Pain	2017-12-05 01:35:42.360	7	Kamal
3	Urgent	2017-12-06 01:03:38.890	7	Rahim
4	Come Soon	2017-12-06 10:47:20.980	7	Abul
5	Need Help	2017-12-06 12:08:46.727	7	Dhuly
6	Dear Nurse...please come soon	2017-12-06 15:45:12.840	7	Jamal
7	Hello Nurse....Please Come Soon...	2017-12-07 11:56:35.053	8	Alam

Figure 5.1.3: Call Nurse

A Diagnosis Table Contains the Test Report with Time as bellow:

DiagnosisID	PatientID	DiagnosisReport	Time	NurseID	PatientName
1	7	Creatinine: 9.1	2017-01-01 00:00:00.000	2	Jamal
2	8	Hemoglobin=10.5; RBC=1000;	2017-12-07 12:41:43.960	10	Alam
3	7	TSS=1.5	2017-12-07 12:41:43.960	2	Jamal
NULL	NULL	NULL	NULL	NULL	NULL

Figure 5.1.4: Diagnosis

Prescription Table contains the Prescription details with doctor's name.

PrescriptionID	PatientID	DoctorID	Description	Time	PatientName
1	7	1	Napa= 1+1+1	2017-01-01 00:00:00.000	Jamal
2	8	9	Renitid= 1+0+1; Ace=1+1+1	2017-12-07 14:24:13.573	Alam
3	7	1	Fixo=1+0+1	2017-12-07 14:24:13.573	Kamal
NULL	NULL	NULL	NULL	NULL	NULL

Figure 5.1.5: Prescription

## 5.2 Implementation of Front-end Design

Here we try to describe the total system that we develop:

### Home page and Login:

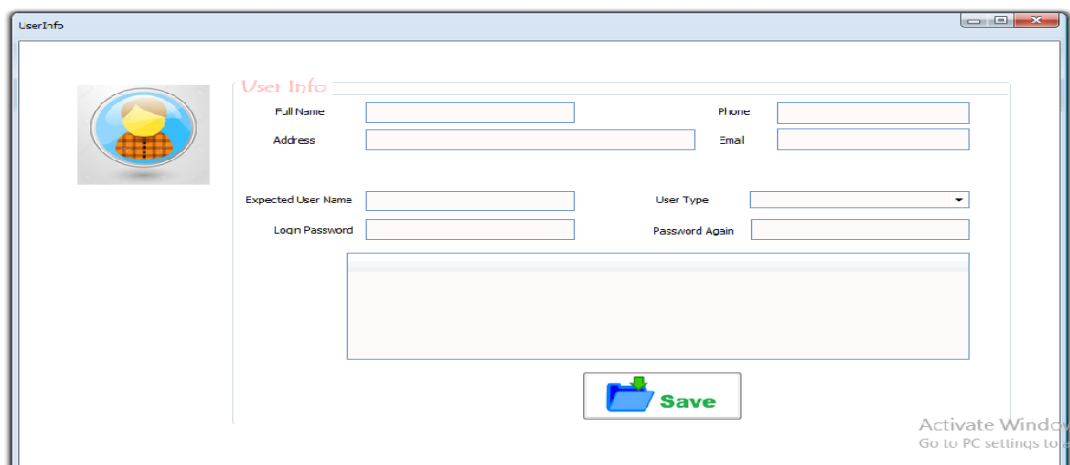
First of all, if any one go to our application, he/she can see the login page first. The following Figure 5.2.1: Homepage.



The screenshot shows the 'Online Patient Service' homepage. At the top, the title 'Online Patient Service' is displayed in a large, red, serif font. Below the title, there is a 'Create New User' button. Underneath this button is a 'Login' section. The 'Login' section contains two input fields: 'User ID:' and 'Password:'. Below these fields is a 'Login' button with a green circular icon and a purple arrow pointing to the right. The entire 'Login' section is enclosed in a light blue border.

Figure 5.2.1: Homepage

Anyone can get detail information from any part of our application. But for putting any information into our application he/she need to conform Sign in and enter our system with valid Email Address and Password. The following Figure 5.2.2 has been user info/sign.



The screenshot shows the 'User Info' form. The form is titled 'User Info' in red. It contains several input fields: 'Full Name', 'Address', 'Phone', 'Email', 'Expected User Name', 'User Type' (a dropdown menu), 'Login Password', and 'Password Again'. There is a large empty text area below the 'Login Password' and 'Password Again' fields. At the bottom of the form is a 'Save' button with a green folder icon. The form is enclosed in a light blue border. In the bottom right corner, there is a small text box that says 'Activate Windows Go to PC settings to activate Windows'.

Figure 5.2.2: user info/sign

## 5.3 Create user

For using this system in the admin level, a user must be created or registered. To complete this task a general blank form needs to be filled with user information, user ID and password. A user can modify the entered data or information about them with the help of administration later. Administration has all the rights to delete any user and create them.

## 5.4 Doctor Portal

A doctor can view Patient Call List and can create Prescription and Diet Plan for any patient. He/ she can also view Diagnosis Report, Bills and Nurse Observation Report. The following Figure :5.4.1 Doctor Portal.

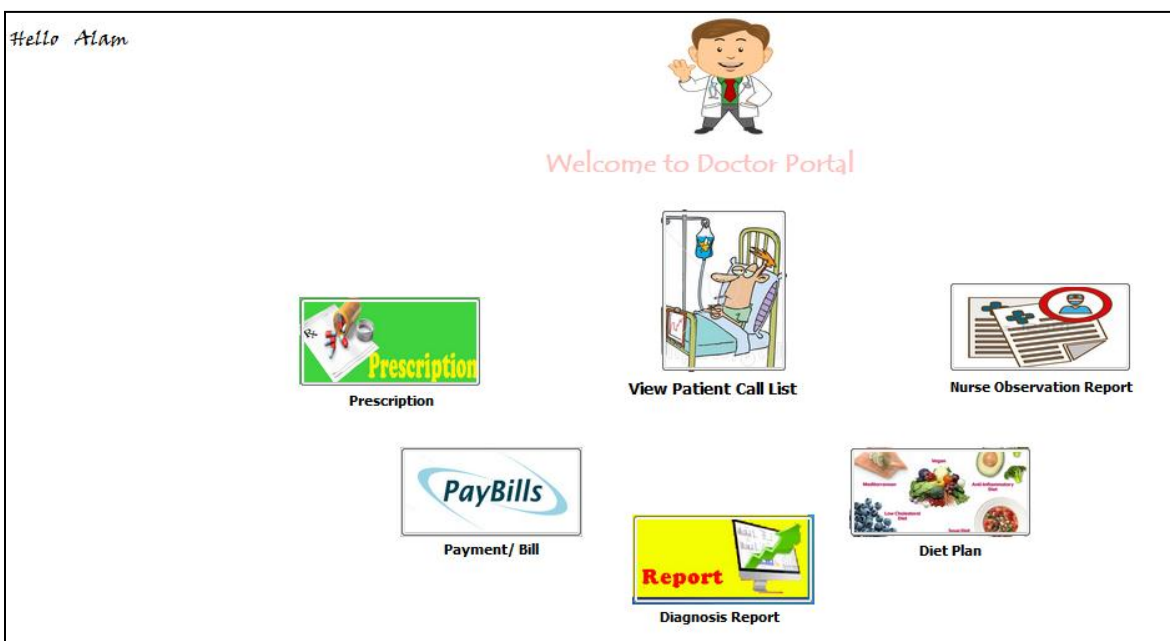


Figure :5.4.1 Doctor Portal

## 5.5 Nurse Portal:

A Nurse can view Nurse Call List and can create Diagnosis Report, Bills and Nurse Observation Report for any patient. He/ she can also Prescription and Diet Plan. The following Figure 5.5.1 has been Nurse Portal.

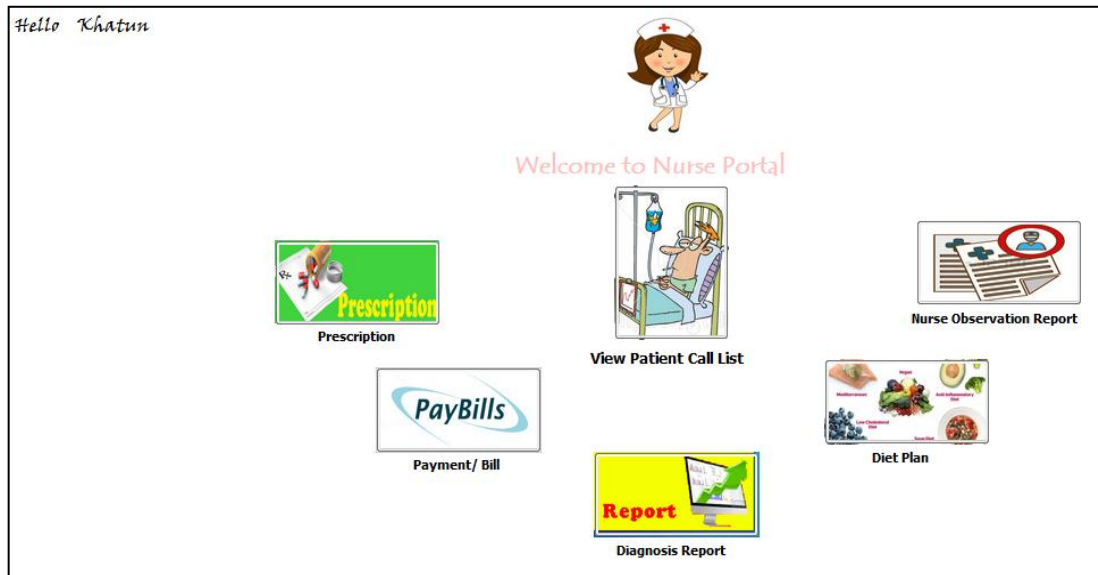


Figure 5.5.1: Nurse Portal

## 5.6 Patient Portal:

A Patient can Call Doctor and Nurse at any time and can view Prescription and Diet Plan, Diagnosis Report, Bills and Nurse Observation Report for any patient. The following Figure 5.6.1 has been Patient Portal.

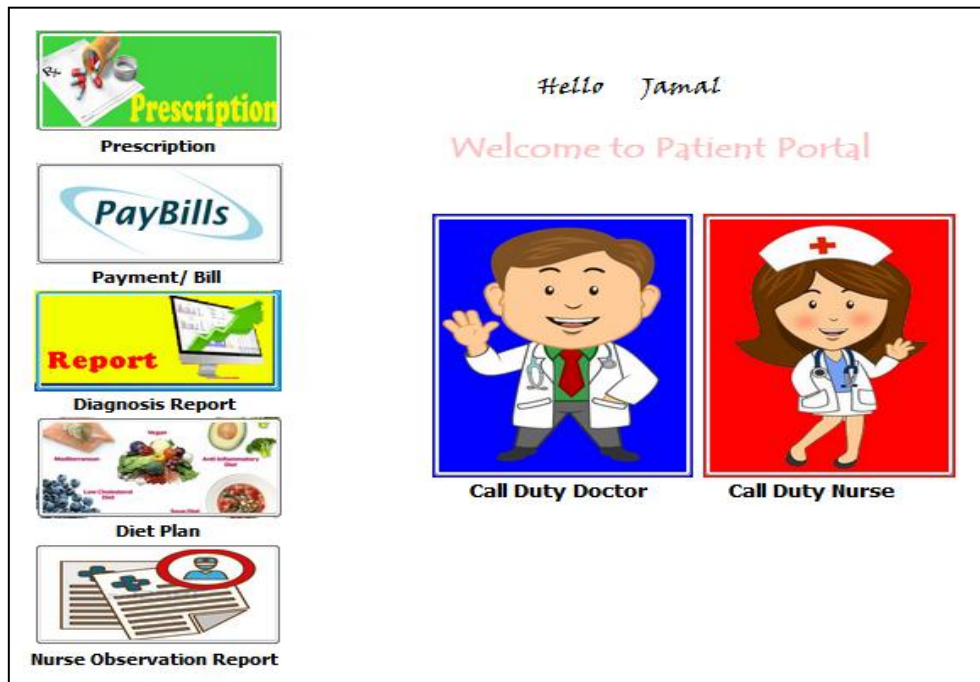


Figure 5.6.1: Patient Portal

### **5.3 Implementation of Interactions**

We implement all interactions like Add a Medicine's information, Add Problems of the patients, Add Pharmacy info, Problem based Medicine search, Select Pharmacy, Pharmacy based Medicine search, Select Medicine, Apply for Order, Confirm Order through email to patient accordingly.

#### **Connection with the Database**

We use "Sql Connection" to connect with Database named "Questions".

To connect with Database, the system need to know following things:

Server Name (Name of the Server PC) = 'DESKTOP-LTPE1RM'

Database Name: Name of the Database= "Medicine"

User of the Database= "sa"

Password of the Database User sa= "123"

In C# we use and Class named Connections which contains following info

### **5.4 Testing Implementation**

We test our solution accordingly. First of all we creates some Test case before developing the system. Followings are some samples of Test Cases:

Test Case 1: Patient Service

In our system, a patient can do: Call for a doctor, Call for Nurse, View Prescription, View Diagnosis Report, View Diet Chart, View Bills, and View Patient Observation. We can Test to Call a Doctor by a Patient with following Test Data:

1. Doctor Name: Dr. Wakhil
2. Problem: Pain and Fever
3. Time: 12:30 PM
4. Patient Name: Mr. Jamal



## **Test Case 2: Doctor Service**

In our system, a doctor can do: Prescription, Diet Chart, View Call List, and Call for a Nurse, View Diagnoses Report, and View Patient Observation Report. A Doctor can write Prescription with following Test Data:

1. Patient Name: Mr. Jamal
2. Description: Napa=1+0+1, Renited=1+0+0
3. Doctor Name: Dr. Wakhil

## **Test Case 3: Nurse Service**

In our system, a Nurse can do: View Call List, Patient Observation Report, Diagnoses Report Entry, Bill Entry, and Call for a Doctor. A Nurse can write on Observation Report with following Test Data:

1. Patient Name: Mr. Jamal
2. Observation: Temperature: 102, Blood Presser: 180/90
3. Nurse Name: Ms. Alom Ara

## **5.5 Test Results and Reports**

After developing the system, we found all Test Cases are passed successfully.

- Test case 1: Passed Successfully
- Test case 2: Passed Successfully
- Test case 3: Passed Successfully

## **CHAPTER 6**

### **Conclusion and Future Scope**

#### **6.1 Discussion and Conclusion:**

Actually, we have a goal to support all the patients of any hospital in all the ways so that he/ she can get emergency service at any time. Our service will help a patient a lot and save their life. Doctor and Hospital authority also can provide better service to their patient.

Limitations:

There is some limitation of our system also which are as bellow:

1. Our system cannot send email to the concern people
2. Need a conversation panel with the contact person and others

#### **6.2 Scope for Further Developments:**

We will incorporate following things in future:

1. Notification through email
2. Notification through SMS (SMS Charge applicable)
3. Introduce Conversation Panel

In conclusion, we can say, our motive is to help a patient through our web application and not doing any business from here. We try to realize the needs of the patients and problems of the patient. And try to provide support accordingly. We hope and expect more suggestion from the patients to make it successful and used application to all.

## REFERENCES

**There are no sources in the current document.**

[1] “Programming guide,” available online <https://docs.microsoft.com/en-us/dotnet/csharp/programming-guide/>. [Last Access on 27 May 2019].

[2] “language section,” available online <https://docs.microsoft.com/en-us/dotnet/csharp/getting-started/introduction-to-the-csharp-language-and-the-net-framework> [Last Access on 27 May 2019]

[3] “Library,” [https://msdn.microsoft.com/en-us/library/aa288436\(v=vs.71\).aspx](https://msdn.microsoft.com/en-us/library/aa288436(v=vs.71).aspx) [ Last Accessed 15 June 2019]

[4] “Framework,” <https://docs.microsoft.com/en-us/dotnet/framework/> [ Last Accessed 18 June 2019]

[5] “Design and Development,” <https://visualwebgui.codeplex.com> [ Last Accessed 22 July 2019]

[6] “Weblogs” <https://weblogs.asp.net/visualwebgui/visual-webgui-on-msdn>

<http://www.crystalreports.com> [Last Accessed 2<sup>nd</sup> August 2019]

[7] “Implementation,” <https://docs.microsoft.com/en-us/dotnet/csharp/tutorials/> [Last Accessed on 28 August 2019]

[8] “Tutorial for sql,” <https://docs.microsoft.com/en-us/sql/sql-server/tutorials-for-sql-server-2016> [ Last Accessed 4th September 2019]

[9] “Related Work”, <https://selecthub.com/medical-software/patient-management-software-features/> [ Last Accessed 10th September 2019]

[10] [https://en.wikipedia.org/wiki/Patient\\_management\\_software](https://en.wikipedia.org/wiki/Patient_management_software) [Last Accessed 12th September 2019]

### **Project Drive Link:**

<https://sites.google.com/diu.edu.bd/patient>

## APPENDIX

### Connection String Source Code:

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Data.SqlClient;

namespace HRMSys
{
    public class Connectivity
    {
        private SqlConnection connection;
        public SqlConnection Connection
        {
            get { return this.connection; }
        }
        public void GetConnection()
        {
            string source = "server=DESKTOP-CUEHIQ1\\SQLEXPRESS; Persist Security Info=true;
user=sa; password=start777; initial Catalog=patient";
            connection = new SqlConnection(source);
            connection.Open();
        }
        public void CloseConn()
        {
            connection.Close();
        }
    }
}
```

### Check Login-Source Code:

```
public void checkLogin()
{
    //Open Connection
    Connectivity conn = new Connectivity();
    conn.GetConnection();
    string selectString = "select UserID, UserName, Type from UserInfo where ExpectedUserID=" +
txtUserID.Text + " and Password =" + txtPassword.Text + """;
    SqlCommand commend = new SqlCommand(selectString, conn.Connection);
    DataTable dt = new DataTable();
    SqlDataAdapter adtp = new SqlDataAdapter(commend);
    adtp.Fill(dt);
    if (dt.Rows.Count > 0)
    {
        Session["sUserId"] = dt.Rows[0][0].ToString();
    }
}
```

```

Session["sUserName"] = dt.Rows[0][1].ToString();
Session["sType"] = dt.Rows[0][2].ToString();
Session["lbldn"] = "";
//Close the Data Reader & Database Connection
clearall();
if (Session["sType"].ToString()=="Doctor")
{
    MainFormDoctor hp1 = new MainFormDoctor();
    hp1.Show();
}
else if (Session["sType"].ToString() == "Patient")
{
    MainFormPatient hp1 = new MainFormPatient();
    hp1.Show();
}
else if (Session["sType"].ToString() == "Nurse")
{
    MainFormNurse hp1 = new MainFormNurse();
    hp1.Show();
}
}
else
{
    MessageBox.Show("User ID or Password is not Matched...");
}
conn.CloseConn();
}

```

### **Save User Info-Source Code**

```

public void saveUserInfo()
{
    //Open Connection
    Connectivity conn = new Connectivity();
    conn.GetConnection();

    //string Pass;

    if (txtPass1.Text == txtPass2.Text)
    {
        int n;
        bool isName = Regex.IsMatch(txtName.Text, @"^[a-zA-Z-]+$"); //LetterDash
        bool isNumericPhone = int.TryParse(txtPhone.Text, out n);
        bool email = IsValidEmail(txtEmail.Text);

        if (isName)
        {
            if (isNumericPhone)
            {

```

```

        if (email)
        {

            string insertString = "insert into UserInfo (UserName, Address, Phone, Email, Type,
ExpectedUserID, Password) values ('" + txtName.Text + "', '" + txtAddress.Text + "', '" + txtPhone.Text
+ "', '" + txtEmail.Text + "', '" + cmdType.Text + "', '" + txtExpectedUser.Text + "', '" + txtPass1.Text +
"' )";

            //execute SQL statement
            SqlCommand commend = new SqlCommand(insertString, conn.Connection);
            SqlDataReader reader = commend.ExecuteReader();

            //Close the Data Reader & Database Connection
            reader.Close();
            reader.Dispose();
            conn.CloseConn();

            //Clear all after Saving Data
            ClearText();

            //Message
            MessageBox.Show("Saved Succesfully...!");

            Load();
        }
        else
        {
            MessageBox.Show("Invalid Email...!");
        }
    }
    else
    {
        MessageBox.Show("Invalid Mobile No...!");
    }
}
else
{
    MessageBox.Show("Invalid Full Name (Not Contain any Number or special characters)...!");
}
}
else
{
    MessageBox.Show("Password are not matched...!");
}
}
}

```

### **Call Doctor-Source Code**

```

public void saveCallDoctor()

```

```

{
    //Open Connection

    Connectivity conn = new Connectivity();

    conn.GetConnection();

    //insert SQL statement

    //container is a string variable

    //string insertString = "insert into ItemInfo (ItemName, ItemBrand, ItemCode, ItemDescription,
    ItemGroupId ) values (" + txtItemName.Text + ", " + txtItemBrand.Text + ", " + txtItemCode.Text + "
    , " + txtItemDescription.Text + ", " + cmbItemGroup.SelectedValue.ToString() + ")";

    string insertString = "insert into CallDoctor (Problem, Time, PatientID, PatientName ) values (" +
    rtxtProblem.Text + ", getdate() , " + Session["sUserId"] + ", " + Session["sUserName"] + ")";

    //execute SQL statement

    SqlCommand commend = new SqlCommand(insertString, conn.Connection);

    SqlDataReader reader = commend.ExecuteReader();

    //Close the Data Reader & Database Connection

    reader.Close();

    reader.Dispose();

    conn.CloseConn();

    MessageBox.Show("Called Succesfully...!");
}

```

### **Call Nurse**

```

public void saveCallNurse()
{
    Connectivity conn = new Connectivity();

    conn.GetConnection();

    //string insertString = "insert into ItemInfo (ItemName, ItemBrand, ItemCode, ItemDescription,
    ItemGroupId ) values (" + txtItemName.Text + ", " + txtItemBrand.Text + ", " + txtItemCode.Text + "
    , " + txtItemDescription.Text + ", " + cmbItemGroup.SelectedValue.ToString() + ")";

```

```
string insertString = "insert into CallNurse (Problem, Time, PatientID, PatientName ) values ('" +  
rtxtProblem.Text + "', getdate() , '" + Session["sUserId"] + "', '" + Session["sUserName"] + "')";  
  
//execute SQL statement  
  
SqlCommand commend = new SqlCommand(insertString, conn.Connection);  
  
SqlDataReader reader = commend.ExecuteReader();  
  
reader.Close();  
  
reader.Dispose();  
  
conn.CloseConn();  
  
MessageBox.Show("Called Succesfully...!");  
  
}
```