

DESIGNING AND DEVELOPMENT OF AN E-PRESCRIPTION SYSTEM

BY

Nasir Ahmed Bhuiyan
ID: 101-15-954

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

Supervised By

Anisur Rahman
Assistant Professor
Department of CSE
Daffodil International University



DAFFODIL INTERNATIONAL UNIVERSITY, DHAKA
BANGLADESH
OCTOBER 21, 2012

APPROVAL

This Project/Internship titled “**Designing and Development of an E-Prescription System**”, submitted by Nasir Ahmed Bhuiyan 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 October 21, 2012.

BOARD OF EXAMINERS

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

Dr Yousuf Mahbubul Islam Internal Examiner
Professor
Department of Computer Science and Engineering
Faculty of Science & Information Technology
Daffodil International University

Dr Md Kabirul Islam Internal Examiner
Associate Professor
Department of Computer Science and Engineering
Faculty of Science & Information Technology
Daffodil International University

Dr Muhammad Shorif Uddin External Examiner

Professor

Department of Computer Science and Engineering
Jahangirnagar University

DECLARATION

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

Supervised by:

Anisur Rahman
Assistant Professor
Department of CSE
Daffodil International University

Submitted by:

(Nasir Ahmed Bhuiyan)
ID: -101-15-954
Department of CSE
Daffodil International University

ACKNOWLEDGEMENT

At first I would like to express my gratitude and heartiest thanks to the omnipotent Allah for his endless blessing which makes me possible to complete this project successfully.

I would like to express my special thanks to **Mr. Anisr Rahman, Assistant Professor**, Department of CSE Daffodil International University, Dhaka for his guidance, inspiration and constructive suggestions that helped me in the preparation of this project. His tolerated mentality, continual and friendly supervision, constructive criticism, valuable suggestion, careful looks over on all the drafts and correction at all the stages have made possible to complete this project.

I would like to express my heartiest gratitude to Md. Mahmudul Hasan Raju, Leturer, Department of CSE and Head, Department of CSE, for their kind help to complete my project and also to other faculty member and the staff of CSE department of Daffodil International University.

Finally, I would like to thank my family especially my parents and my elder brother for their continuous supports and encouragements throughout the whole period of this project.

ABSTRACT

Bangladesh is a small country with a lot of probabilities. Besides many other sectors our health care sector is improving day by day. Though health care facility of the country is not adequate according to its demand, the government is sincere to improve the present condition. . I think this small step, providing a computer based patient information recording and preserving system will contribute to minimize the problem in the health sector of the country.

The purpose of the E-Prescription system is to serve the people by providing computer based information recording system and allowing a complete prescribing facility to the physicians or the doctors with the related options of medical terminology. The authority or the doctors from another location or from the remote location will be able to retrieve the information provided by the patients through ensuring proper authentication from a central database.

Microsoft visual studio 2008 is used here for designing the forms and implementing the techniques.

After implementation it is tested in different ways and the output is found in desired format.

TABLE OF CONTENTS

CONTENS PAGE

Board of examiners i

Declaration ii

Acknowledgements iii

Abstract iv

CHAPTER

CHAPTER 1: Introduction 1-3

1.1 Purpose 2

1.2 Scope 2

1.3 Tools & Technologies Used 2

1.4 Overview 3

CHAPTER 2: Requirement Analysis 4-7

2.1 Introduction	5	
2.2 Stakeholder Identification		5
2.3 Stakeholder Interviews	5	
2.4 Use Cases		5
2.4.1 Use Case Diagram of Patient	6	
2.4.2 Use Case Diagram of Receptionist		6
2.4.3 Use Case Diagram of Doctor		7
CHAPTER 3: Requirement Specification		8-14
3.1 Overall Description		9
3.1.1 Product Perspective		9
3.1.1.1 System Interfaces	9	
3.1.1.2 User Interfaces	9	
3.1.1.3 Hardware Interfaces	9	
3.1.1.4 Software Interfaces	10	
3.1.1.5 Communication Interfaces		11
3.1.1.6 Memory Constraints		11
3.1.1.7 Operations		11

3.1.2 User Characteristics	11
3.1.3 Constraints	12
3.1.4 Definition of Problem	12
3.1.4.1 Lack of immediate retrievals	12
3.1.4.2 Lack of immediate information storage	12
3.1.4.3 Lack of prompt updating	12
3.1.4.4 Error prone manual calculation	13
3.1.4.5 Preparation of accurate and prompt reports	13
3.2 Specific System Requirements	13
3.2.1 External Interface Requirements	13 13
3.2.2 Software System Attributes	

3.2.2.1 Planned approach towards working	13
3.2.2.2 Accuracy	13
3.2.2.3 Reliability	13
3.2.2.4 No Redundancy	14
3.2.2.5 Immediate retrieval of information	14
3.2.2.6. Immediate storage of information	14 14
3.2.2.7 Easy to Operate	14
3.2.3 Project Hardware Requirement	

15-17

Chapter 4: Methodology of the Proposed System

4.1 Introduction	16	16
4.2 Requirement Analysis and Specification	17	
4.3 Planning	17	17
4.4 Design	17	
4.5 Implementation	17	
4.6 Integration and Testing		

18-42

Chapter 5: Development of the Proposed System

5.1 Introduction	19
5.2 DFD	19
5.3 Process Flow Diagram	20
5.4 Table	21
5.4.1 Patient table	22
5.4.2 User Table	22

5.5 Forms	
22	
5.5.1 Main Form	22
5.5.2 Main Form after Receptionist Login	23
5.5.3 Patient Registration at receptionist desk	23
5.5.4 Token providing to the patient	24
5.5.5 Providing printed token to the patient	24
5.5.6 Doctor's log in	25
5.5.7 Retrieving registration data for the patient	25
5.5.8 Assigning medicine, test next meeting date and prescribe	26
5.5.9 Prescribing and providing the printout to the patient	26
5.6 Codes	
5.6.1 Code for the MDI main form	27-30
5.6.2 Code for the Log In form	31-33
5.6.4 Code for the Prescription Form	33-42
Chapter 6: Conclusion	43-44
6.1 Conclusion	44
6.2 Future Works	44
REFERENCES	45-46

LIST OF FIGURES

FIGURES	PAGE NO
Figure 2.1: Use Case Diagram of Patients	6
Figure 2.2: Use Case Diagram of Receptionist	6
Figure 2.3: Use Case Diagram of Doctor	7
Figure 4.1: Rapid Prototype Software Life Cycle Model	16
Figure 5.1: DFD	19
Figure 5.2: Process Flow Diagram	20
Figure 5.3: Main Form	22
Figure 5.4: Main Form After Login	23
Figure 5.5: Patient Registration Form	23
Figure 5.6: Token Form	24
Figure 5.7: Generated Token	24
Figure 5.8: Doctor's Login	25
Figure 5.9: Prescription Form	25
Figure 5.10: Prescribing	26
Figure 5.11: Generated Prescription	27

LIST OF TABLES

TABLE	PAGE NO
Table 5.1: Patient Table	21
Table 5.2: User Table	22

Chapter 1
Introduction

1.1 Purpose

The purpose of the project is to computerize the conventional prescribing process of hospitals and to develop software which is user friendly, simple, fast, and cost-effective. It deals with the collection of patient's information, diagnosis details, etc. Traditionally, these are done manually. The main function of the system is to register and store patient's details and retrieve these details when required and also to manipulate these details meaningfully. System input contains patient details, diagnosis details.

1.2 Scope

This system can be used in any Hospital, Clinic, Pharmacy or Pathology labs for maintaining patient details and their test results.

1.3 Tools & Technologies used

This project will be a desktop application to be developed in C# having MySQL Database as backend.

Database Design (MySQL Database using SQLyog)

Database Remote Serve (XAMPP)

Form Designing and Coding (C#)

Reporting Tool (Microsoft Reporting)

1.4 Overview

The project maintains two types of users. They are-

- Doctor to prescribe and monitor.
- Receptionist to patient Registration and token providing.

Main opportunities available in this project are-

- Maintaining records of patients.
- Maintaining patients diagnosis details, advised tests to be done.
- Providing different test opportunities to a doctor for diagnosis of patients.
- An integrated mail sending module to send the daily report to the higher authority.

In this project collection of data is from different pathology labs. Results of tests, prescription, precautions and diet advice will be automatically updated in the database. Related test reports, patient details report, prescription and billing reports can be generated as per user requirements. User or Administrator can search a patient's record by his/her unique Id number.

Chapter 2
Requirement Analysis

2.1 Introduction

Electronic Prescription is a form of Computerized Physician Order Entry (CPOE). Contrary to the traditional method of prescribing with the use of paper, an electronic prescribing system allows for the registration and consultation of medication information by medical specialists and other health care providers. The main goal of the project is to reduce the public hassle at hospitals and to ensure the patient information recording system. To achieve this goal we have to go through with some specific course of action. First of all we need to define all requirements to develop the new system in replace of the conventional system.

2.2 Stakeholder Identification

The candidate system is for the betterment of the conventional prescription providing system in the hospital. There are three types of people involved in the system. ○

The Patients

They are the main beneficial of the system.

○ **The Doctors and the Receptionist**

Both of them are the main users or operators of the system, though the doctors have some role of administration.

2.3 Stakeholder Interviews

I have met and discussed with the related stakeholders before developing the system. Have collected data and have decided before design the system. Actually the stakeholder's motives are the vital issue for a candidate system. Even after designing the system I have met with a high official of the Government's health sector to take his suggestion about the system.

2.4 Use Cases

A use case diagram describes the functional requirements for the candidate system. According to the my system the related use case diagram describes the overall process of the system

2.4.1 Use Case Diagram of Patients

Figure 2.1 shows the use case diagram of patients. At first patients come to the receptionist desk and get a printed token after completion of the registration. Then they will go to the particular doctor's room for the particular problems.

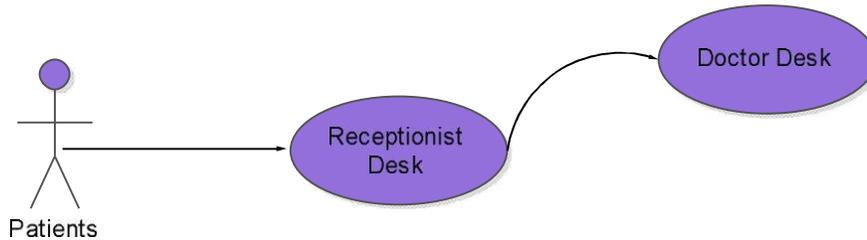


Figure 2.1: Use Case diagram for the patients

2.4.2 Use Case Diagram of Receptionist

Figure 2.2 shows the use case diagram for the receptionist. At first the receptionist will log in through their pc by providing the specific "id" and "password" and enter the information of the patients in the registration form. After completion of the registration they will be able to print the token of that patient even they will be able to print the token for an old patient using the patient's id.

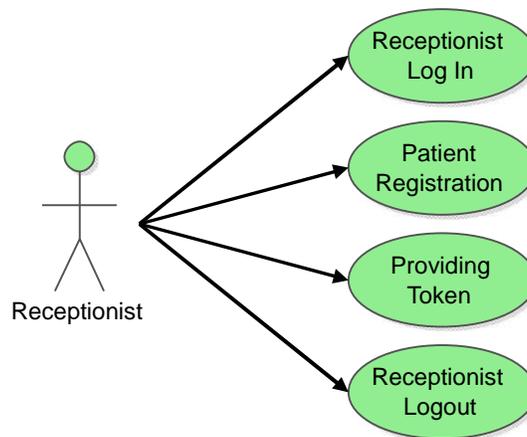


Figure 2.2: Use Case diagram for the Receptionist

2.4.3 Use Case Diagram of Doctor

Like receptionist the doctor will also log in through his pc and after coming a patient to him, he will show the information of the patients using the patient's id. Then the doctor will prescribe on the prescription form. Finally the doctor will provide a printed prescription to the patient. At the end of the day the doctor will send the daily report to the higher authority from the application by the Daily Report menu. At first he will show the daily report then will save it in Microsoft Excel format finally he will attach the excel file and will mention the recipient id, mail subject and some text in the body part. Finally he will send the mail and a confirmation will be appeared.

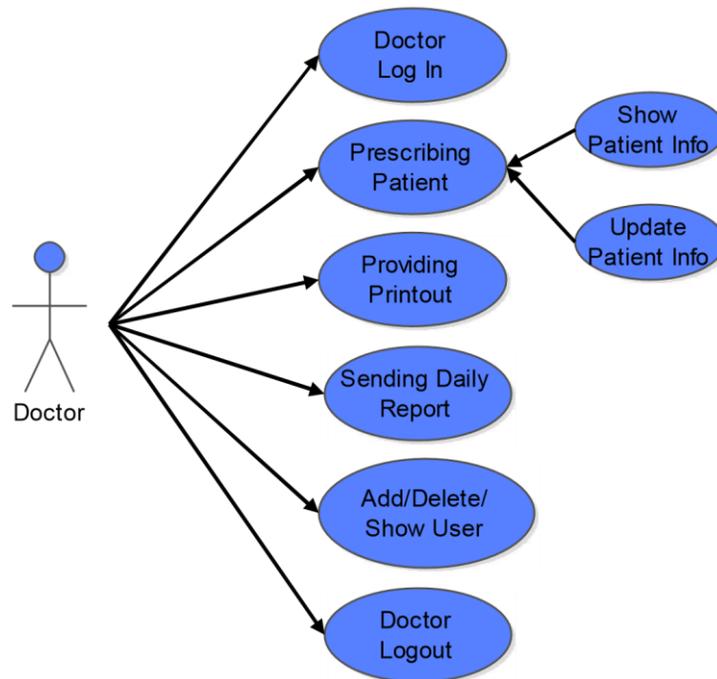


Figure 2.3: Use Case diagram for the Doctors

Chapter 3
Requirement Specification

3.1 Introduction

The Software Requirements Specification (SRS) is the requirements specification for the software system that's the overall description of the system being developed.

3.1 Overall Description

This part will describe the functions and their target in the project. It will also describe the constraints and the technical requirements of the project.

3.1.1 Product Perspective

E-Prescription System will provide a group of action with relative interface environment for the user. There will be a reliable database also to provide the recorded information.

3.1.1.1 System Interfaces

The software is connected with a MySQL based database server. Besides this all the pc's for the receptionist's and the doctor's will be interconnected in a LAN.

3.1.1.2 User Interfaces

The software is a desktop based application. The form has been design as per as user friendly. The main form is an enriched MDI form. Same Login form for all user. After log in, every user will get the same window with distinct menu option as their role and level.

3.1.1.3 Hardware Interfaces

There won't need any special types of hardware interface in the system.

3.1.1.4 Software Interfaces

1. **Name:** Microsoft Visual Studio **Version:** 2008, .Net Framework 3.5

Source: Microsoft Corporation

Purpose: The Microsoft Visual Studio 2008 provide an enriched OOP based software developing language C# to design and code the whole application.

Definition of the Interface: The .Net 3.5 based C# provides a very strong GUI based tools collection to design interactive forms and reports. It has a very enriched IDE to code and debugging tools.

2. **Name:** MySQL

Version: 5.5.16

Source: MySQL

Purpose: To use as database server.

Definition of the Interface: MySQL the world's most popular open source database as they declared in their website. MySQL provides high performance, high reliability and ease of use. MySQL runs on more than 20 platform's including Linux, Windows, Mac OS, Solaris, and IBM AIX. MySQL has become the preferred choice of corporate IT Managers because of its elimination's capability of the major problems associated with downtime, maintenance, administration and support.

3. **Name:** SQLyog Enterprise

Version: 6.56

Source: Webyog software private Ltd.

Purpose: To work with the MySQL in a GUI environment.

Definition of the Interface: SQLyog MySQL GUI is the most powerful MySQL manager and admin tool, combining the features of MySQL Administration, php MyAdmin and other MySQL front ends and MySQL GUI tools.

4. Name: XAMPP

Version: 2.5

Source: Apache Friends

Purpose: To use the MySQL Server in local pc as server platform.

Definition of the Interface: XAMPP is an easy to install apache distribution containing MySQL, php and perl. It is an open source tools and easy to download and get.

3.1.1.5 Communication Interface

The default communication protocol for the data transmission between server and the client pc is the Transmission Control Protocol / Internet Protocol (TCP/IP).

3.1.1.6 Memory Constraints

There is not a specific memory constraint for this application.

3.1.1.7 Operations

The main operations are the information entry on the form, saving them in the database, updating, searching and printing the stored information.

3.1.2 User Characteristics

The user types that are being used in the application are as follows:

Receptionist: They are usually meet with the patients at the beginning then collect the basic information from the patients. According to the patient's chief complain they provide a printed token to the patients with the information of the doctor's room number. They have the ability to register patients, token printing and searching the old patient's information for token providing.

The receptionist's need some basic skill as follows:

- Comfortable of working with computer.
- He/she must have knowledge in medical field.
- He/she must also have basic knowledge of English too. **Doctor:**

Doctors have some administrative capability of adding, searching and deleting user in the application. Their main responsibility is the prescription providing and providing printout to the patients. The doctor needs the adequate knowledge, skills and the experience for the specific department or disease he/she has been appointed for providing treatment.

3.1.3 Constraints

- GUI is only in English.
- Login and password is used for identification of user and there is no facility for guest.

3.1.4 Definition of Problems

Problems in conventional system:

3.1.4.1 Lack of immediate retrievals

The information is very difficult to retrieve and to find particular information like- E.g. - To find out about the patient's history, the user has to go through various registers. These results are inconvenience and wastage of time.

3.1.4.2 Lack of immediate information storage

The information generated by various transactions takes time and efforts to be stored at right place.

3.1.4.3 Lack of prompt updating

Various changes to information like patient details or immunization details of child are difficult to make as paper work is involved.

3.1.4.4 Error prone manual calculation

Manual calculations are error prone and take a lot of time this may result in incorrect information. For example, calculation of patient's arrival in a day for various treatments.

3.1.4.5 Preparation of accurate and prompt reports

This becomes a difficult task as information is difficult to collect from various registers.

3.2 Specific System Requirements

3.2.1 External Interface Requirements

Interfaces external to this software can be explained as – User Interfaces and that's are the different types of forms used in the application. **3.2.2 Software System Attributes**

3.2.2.1 Planned approach towards working

The working in the organization will be well planned and organized. The data will be stored properly in data stores, which will help in retrieval of information as well as its storage.

3.2.2.2 Accuracy

The level of accuracy in the proposed system will be higher. All operation would be done correctly and it ensures that whatever information is coming from the center is accurate.

3.2.2.3 Reliability

The reliability of the proposed system will be high due to the above stated reasons. The reason for the increased reliability of the system is that now there would be proper storage of information.

3.2.2.4 No Redundancy

In the proposed system utmost care would be that no information is repeated anywhere, in storage or otherwise. This would assure economic use of storage space and consistency in the data stored.

3.2.2.5 Immediate retrieval of information

The main objective of proposed system is to provide for a quick and efficient retrieval of information. Any type of information would be available whenever the user requires.

3.2.2.6. Immediate storage of information

In manual system there are many problems to store the largest amount of information.

3.2.2.7 Easy to Operate

The system should be easy to operate and should be such that it can be developed within a short period of time and fit in the limited budget of the user.

3.2.3 Project Hardware Requirement

Processor	: Pentium Dual Core, Minimum 1.6 GHZ or faster
RAM	: 512 MB
Hard Disk	: 40 GB
Operating System	: Windows xp or higher

Chapter 4

Methodology of the Proposed System

4.1 Introduction

The primary objective of the project to identify the problems in the existing system and to provide a computer based automated system by fixing those problems. After successfully completion of the project it will be provide to the public hospitals to use for providing prescription. During the development of the proposed software the “**Rapid Prototype Software life cycle Model**” will be followed. For testing, each module will be tested individually then it will be integrated and finally input output and usability will be verified.

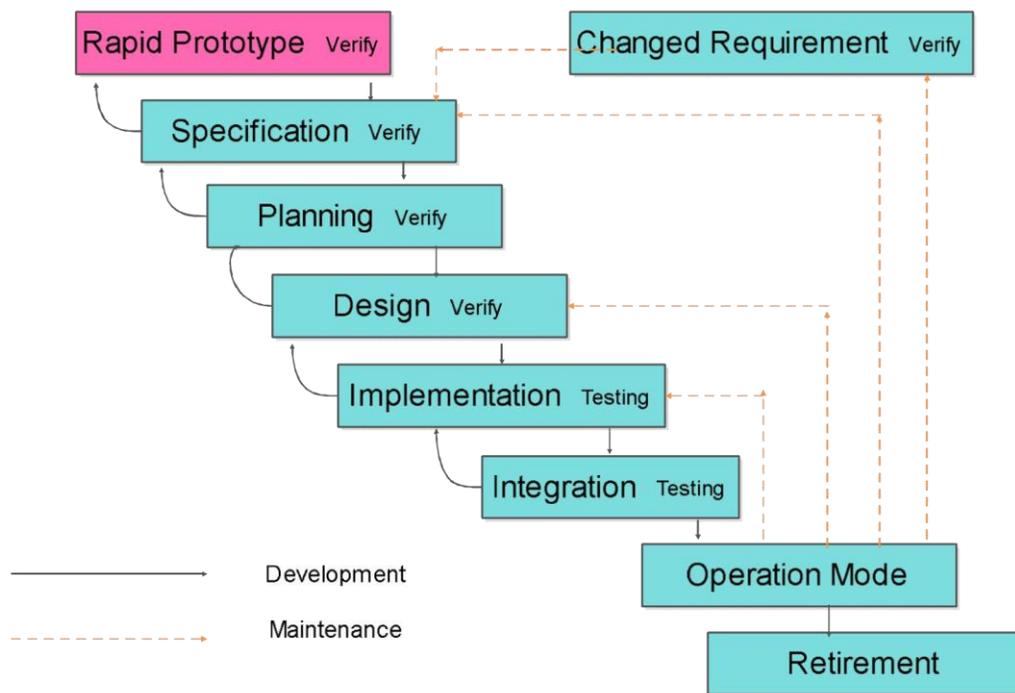


Figure: 4.1 Rapid Prototype Software Life Cycle Model

4.2 Requirement Analysis and Specification

Extracting the requirements and specification of a desired software product is the first task in creating it. While customer probably believe that they know what can the software do and how to do it in the software, then it may require skill and experience in software engineering to recognize incomplete, ambiguous and contradictory requirements. Different hospitals have been visited and studied their prescription

providing methods. The software and hardware requirements are also studied and specified in this phase.

4.3 Planning

A well designed plan is a like a guideline to deliver a high quality software using given resource, timeframe and within the budget available. To develop this software, a will organized plan is used. In planning phase, project organization, quality plan, testing plan and documentation plan is performed.

4.4 Design

Design phase describes desired features and operations in detail, including database design, software design, screen layouts and other documentation. There are different types of design performed to develop this software like, DFD, Process Flow diagram, use case etc.

4.5 Implementation

Implementing the software design into the code and form design is the most significant part of the software. This is the development phase of the application. In this phase codes are written and necessary requirements are assembled to build the software.

4.6 Integration and Testing

In this phase, Brings all the pieces together into a special testing environment, then checks for errors, bugs and interoperability.

Chapter 5
Development of the Proposed System

5.1 Introduction

In order to develop the proposed system different tools are used. The front end user interface is developed by C#. C# is connected with the backend database MySQL Server located in a remote location which has a public IP to get connect through internet.

5.2 DFD

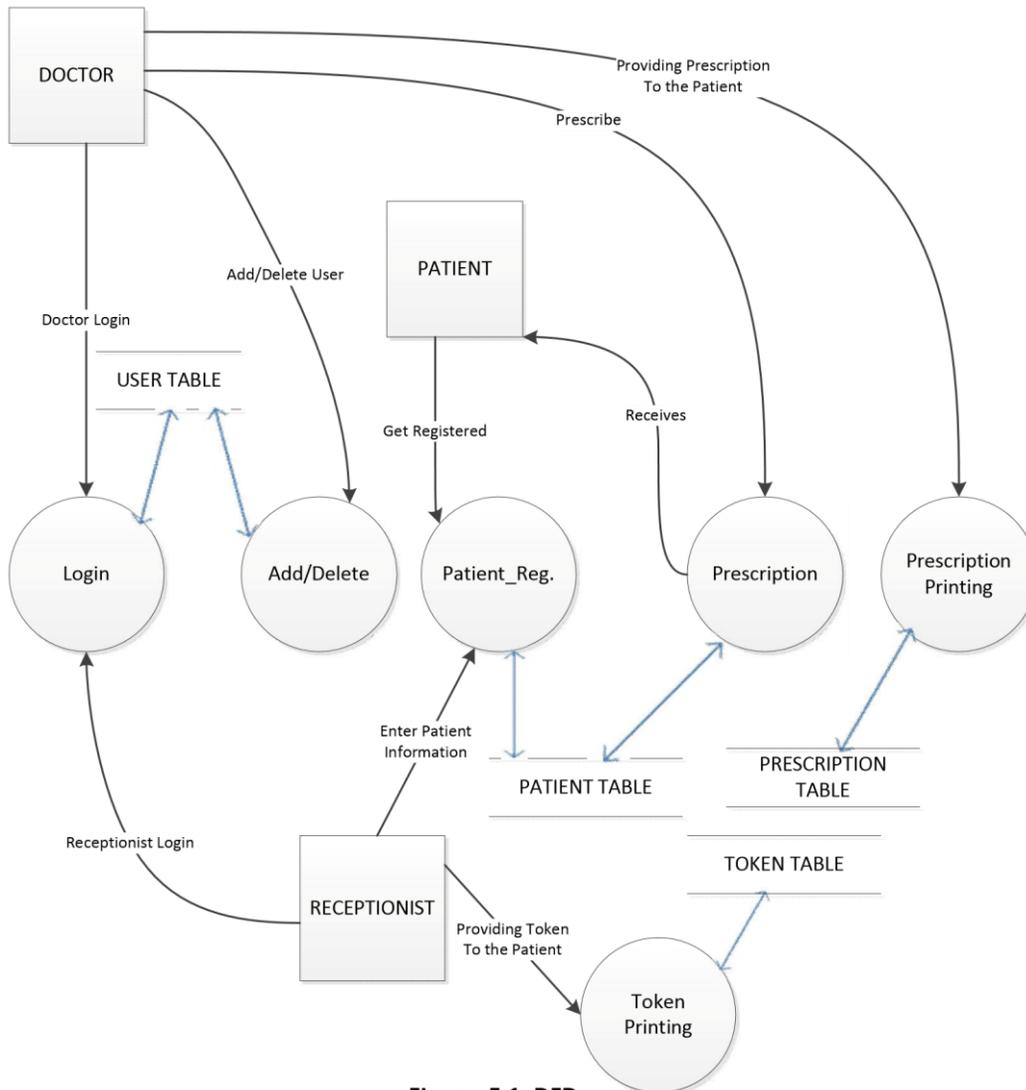


Figure: 5.1, DFD

5.3 Process Flow Diagram

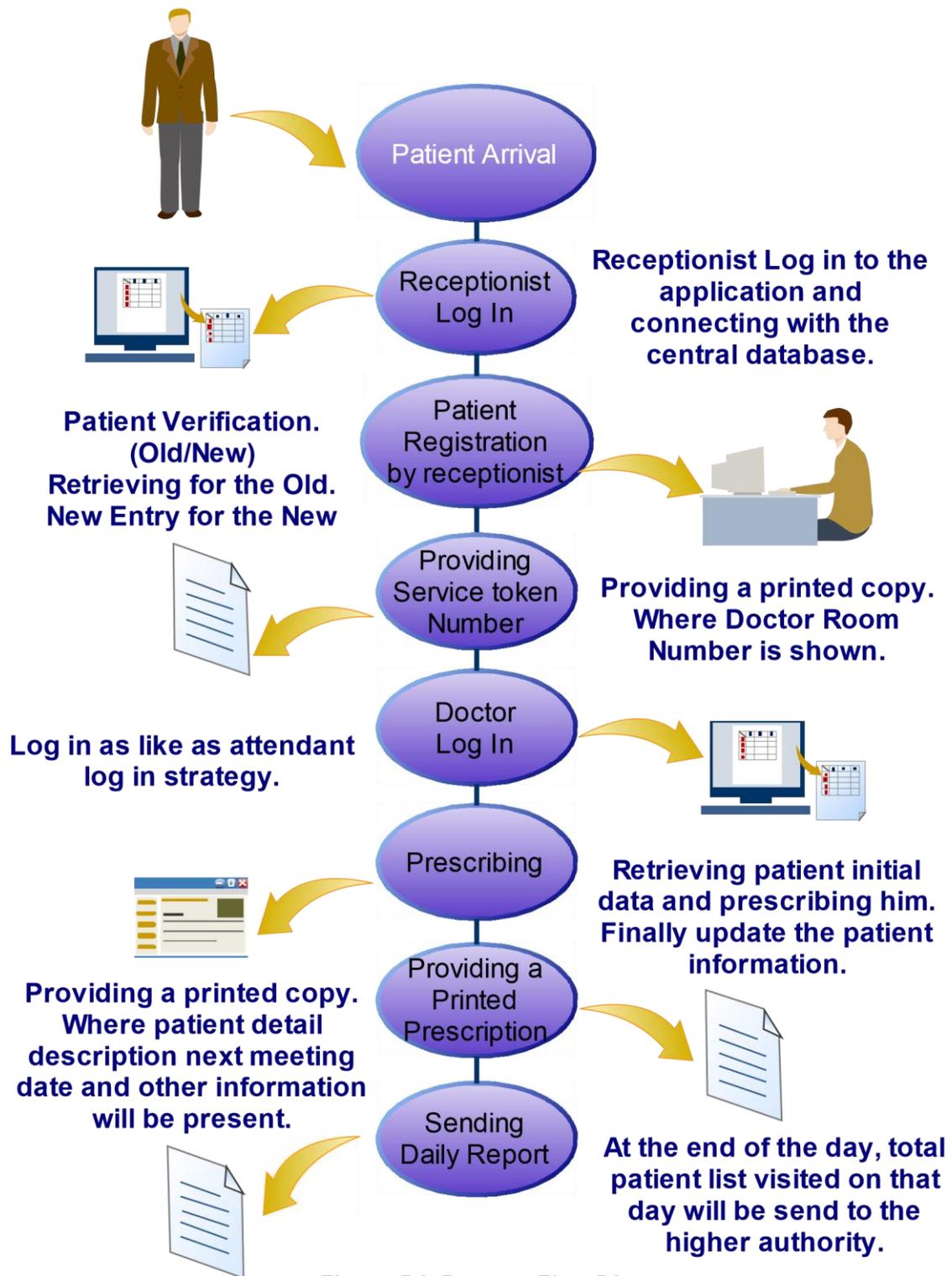


Figure:5.2, Process Flow Diagram

5.4 Table

There are basically two tables used in the application rest of the tables are replica of these two tables used for the report generation.

5.4.1 Patient table

Table 5.1, Patient Table

Fields	Data Types	Constraints	Description
p_Id	int(40)	Primary Key	Patient Id
p_Name	varchar(50)	Not Null	Patient Name
p_Age	int(20)	Not Null	Patient Age
p_Gen	varchar(20)	Not Null	Patient Gender
p_Mstts	varchar(20)	Null	Maried/Non Maried
p_Prblm	varchar(100)	Not Null	Chief Complain
p_Medicine	varchar(25)	Null	Medicine Provided
p_Test	varchar(50)	Null	Suggested Test
p_Prscpn	varchar(200)	Not Null	Prescription
p_Cell	int(20)	Not Null	Cell No
p_Adrs	varchar(100)	Not Null	Address
p_Mdate	date	Null	Meeting Date
p_Cdate	date	Null	Current Date

Patient table is the main table for the prescription. The data entry by the receptionist will be stored in this table. Receptionist will enter in few of the fields, others fields will be filled by the doctor while he will be prescribed.

5.4.2 User Table

Table 5.2, User Table

Fields	Data Type	Constraints	Description
u_Id	int(50)	auto_increment	User Id
u_Name	varchar(50)	Not Null	User Name
u_Pass	varchar(50)	Not Null	User Password
u_Des	varchar(50)	Not Null	Designation
u_Cell	int(15)	Not Null	Cell No
u_Join	date	Null	Joining Date
u_Sal	int(15)	Null	User's Salary

User table contains the application user's information. This table is filled from a form entry by the doctor. Doctor has the ability to insert, delete and show from this table.

5.5 Forms

User will work through some forms in the application. Here are the some basic forms used in the application.

5.5.1 Main Form

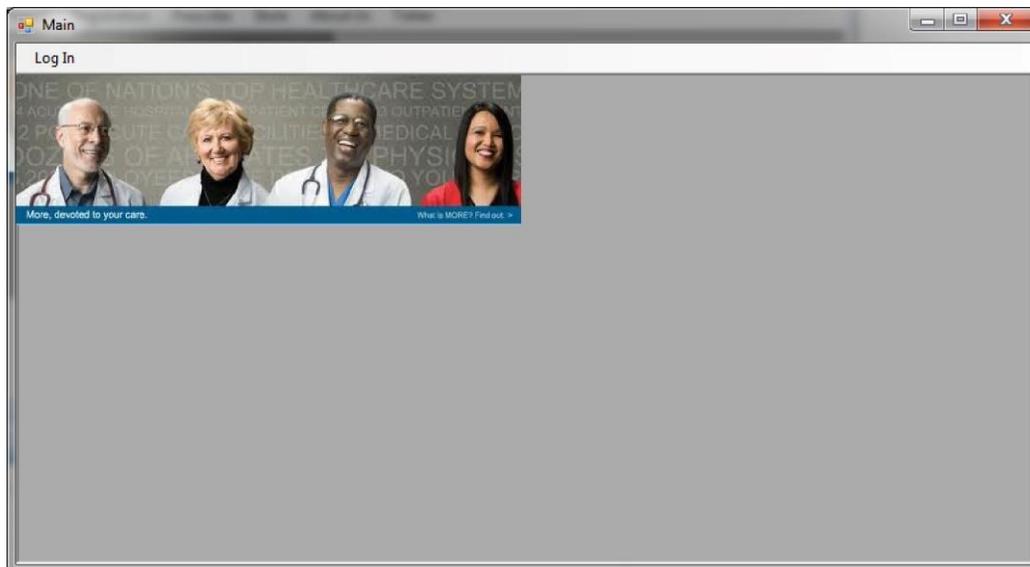


Fig: 5.3, Main Form

Main Form is basically an enriched MDI form which has special features to contain child form and providing menu features.

5.5.2 Main Form after Receptionist Login

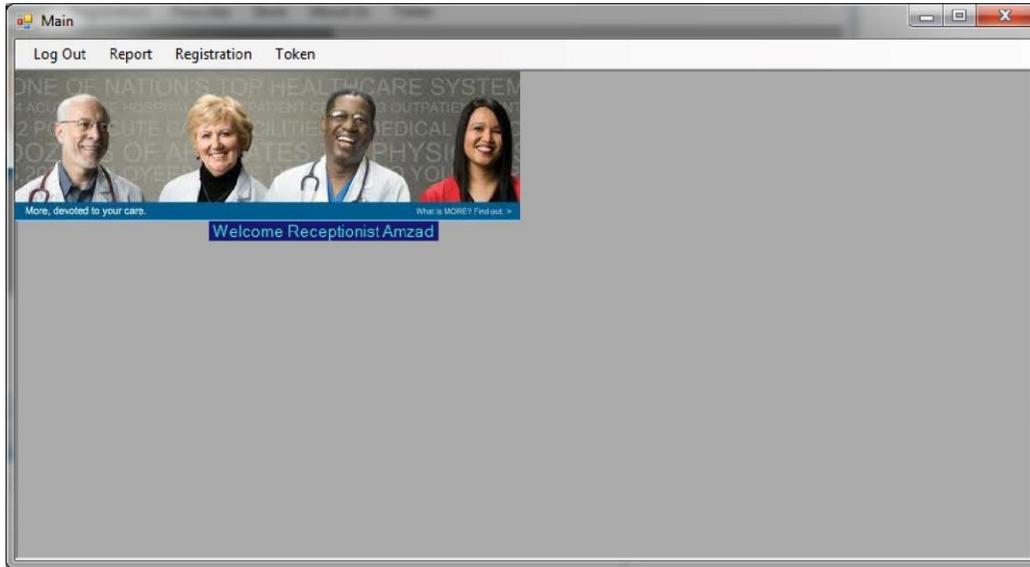


Fig 5.4, Main Form after Login

After log in of a user a label under the banner shows the name and designation of the User.

5.5.3 Patient Registration at receptionist desk

Patient Information	
Gender	Male
Name	Md. Sumon
Marital Status	Un Married
Age	25
Cell No.	01719378240
Problem	Fever
Address	Mohammadpur, Dhaka
Date	Thursday, September 27, 2012

Figure 5.5, Patient Registration Form.

The Main task of the receptionist after log in is the patient registration. He will click on the registration menu to get the registration form, later after collecting the

information from the patient, he will entry those in the form. Finally he will click on “Register” button.

5.5.4 Token providing to the patient

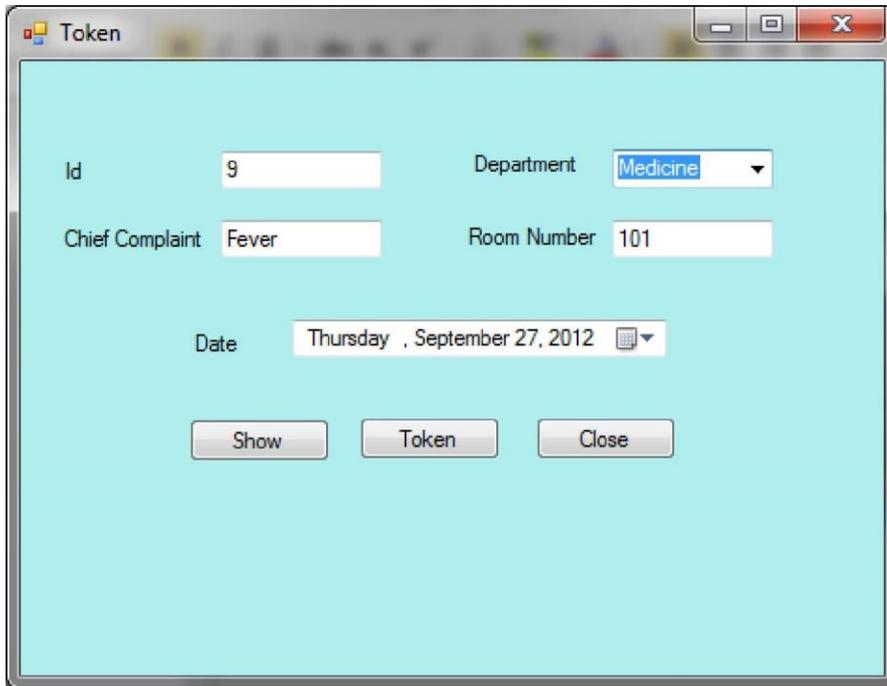
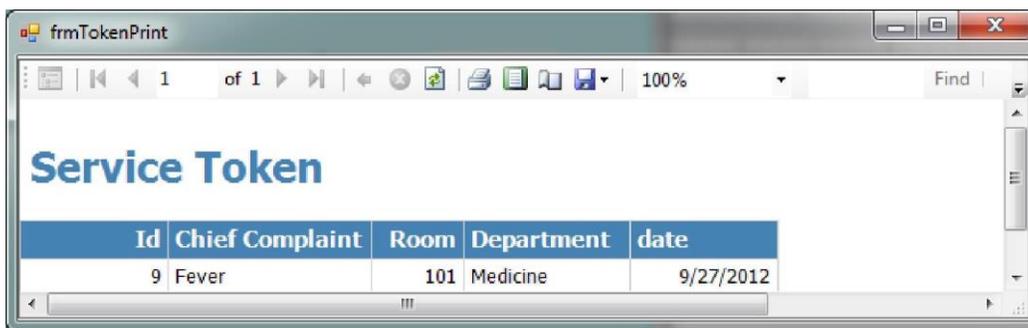


Figure: 5.6, Token Form

To provide a printout token to the patient the receptionist will entry on the form and then will click on “Token Button”.

5.5.5 Providing printed token to the patient



Service Token				
Id	Chief Complaint	Room	Department	date
9	Fever	101	Medicine	9/27/2012

Figure: 5.7, Generated Token

This generated token will be provided to the patient through a print out. The patient will bear the token to the doctor’s room.

5.5.6 Doctor's log in

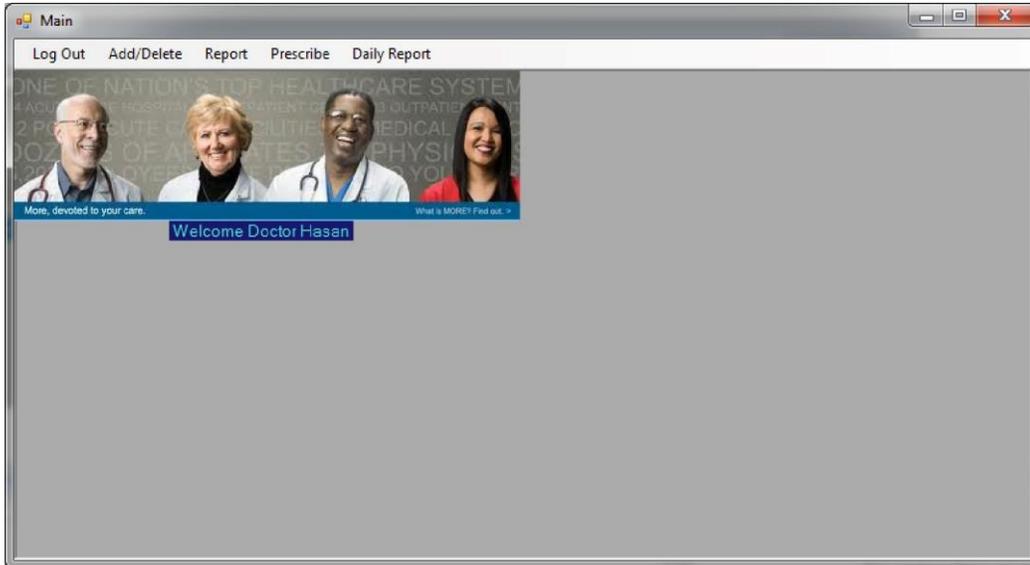


Figure: 5.8, Doctor's Log in.

After a doctor login a label will under the banner will show the doctor's name and designation.

5.5.7 Retrieving registration data for the patient

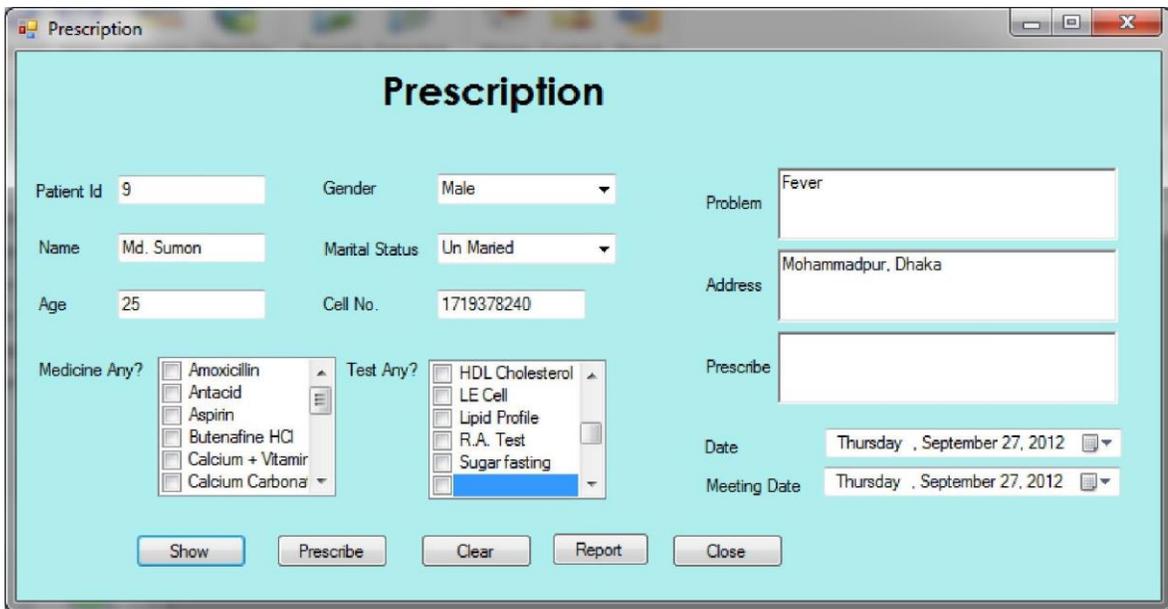


Figure: 5.9, Prescription Form

To provide prescription, at first the doctor will show the patient information by entry the patient's id in the form later he will prescribe the patient.

5.5.8 Assigning medicine, test next meeting date and prescribe



The screenshot shows a software window titled "Prescription" with a light blue background. The form contains the following fields and controls:

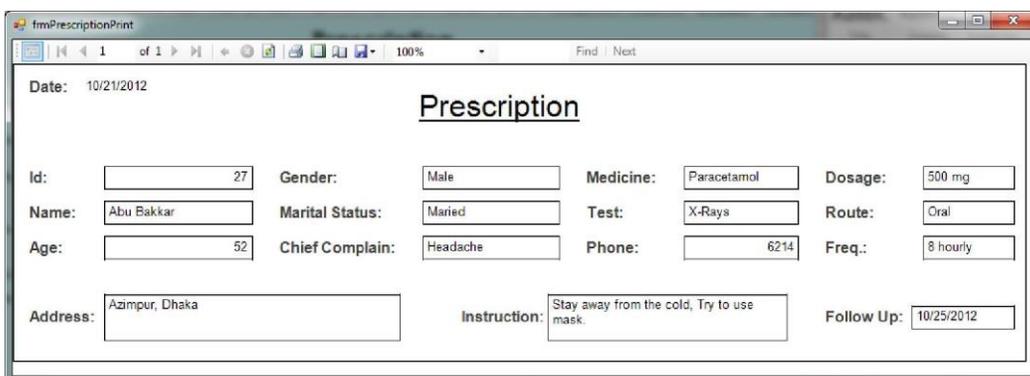
- Patient Information:** Patient Id (27), Gender (Male), Name (Abu Bakkar), Marital Status (Married), Age (52), and Call No. (6214).
- Problem:** Headache
- Address:** Azimpur, Dhaka
- Test Any?:** A list box containing HDL Cholesterol, LE Cell, Lipid Profile, R.A. Test, and Sugar fasting.
- Instruction:** Stay away from the cold, Try to use mask.
- Medicine Any?:** A list box containing Lactulose, Letrozole, Multivitamin Multi, Nitroglycerin, Omeprazole, and Paracetamol.
- Dosage:** 500 mg
- Route:** Oral
- Freq:** 8 hourly
- Date:** 21 October, 2012
- Meeting Date:** 25 October, 2012

At the bottom of the form are five buttons: Show, Prescribe, Clear, Report, and Close.

Figure: 5.10, Prescribing

To prescribe, the doctor will assign te medicine, test and some written instruction as well as meeting date if necessary.

5.5.9 Prescribing and providing the printout to the patient



The screenshot shows a software window titled "fmyPrescriptionPrint" displaying a printed prescription form. The form includes the following information:

- Date:** 10/21/2012
- Id:** 27
- Gender:** Male
- Medicine:** Paracetamol
- Dosage:** 500 mg
- Name:** Abu Bakkar
- Marital Status:** Married
- Test:** X-Rays
- Route:** Oral
- Age:** 52
- Chief Complain:** Headache
- Phone:** 6214
- Freq.:** 8 hourly
- Address:** Azimpur, Dhaka
- Instruction:** Stay away from the cold, Try to use mask.
- Follow Up:** 10/25/2012

Figure: 5.11, Generated prescription.

After completion of the entry the form the doctor will click on “Prescribe” button which will update the patient information then he will click on the “Report” button.

5.6 Codes

5.6.1 Code for the MDI main form

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data; using
System.Drawing; using System.Linq;
using System.Text;
using System.Windows.Forms;

namespace Hospital_Management_System
{ public partial class frmMain :
Form
{
    public frmMain()
    {
        InitializeComponent();
        lblDesg.Visible = false;
        this.logoutToolStripMenuItem.Visible = false;
        this.addDeleteToolStripMenuItem.Visible = false;
        this.reportToolStripMenuItem.Visible = false;
        this.registrationToolStripMenuItem.Visible = false;
        this.prescribeToolStripMenuItem.Visible = false;
        this.storeToolStripMenuItem.Visible = false;
        this.aboutUsToolStripMenuItem.Visible = false;
        //this.tokenPrinToolStripMenuItem.Visible = false;
        this.tokenToolStripMenuItem.Visible = false;

        //this.addDeleteToolStripMenuItem.Visible = false;
    }

    private void toolStripMenuItem1_Click(object sender, EventArgs e)
    {

    }

    private void logInToolStripMenuItem_Click(object sender, EventArgs e)
    {
        //frmLogIn ChildForm = new frmLogIn();        frmLogIn
        ChildForm = new frmLogIn();
        ChildForm.ShowDialog();
        //ChildForm.MdiParent=this;
        //pnlHome.SendToBack();
        //pxrMain.SendToBack();
    }
}
```

```

        //ChildForm.Show();

        if (ChildForm.DialogResult == DialogResult.OK)
        {
            lblDesg.Visible = true;          lblDesg.Text = "Welcome "
+ SystemInfo.designation + " " + SystemInfo.name + "";
            logInToolStripMenuItem.Visible = false;
            logOutToolStripMenuItem.Visible = true;
            toolStripMenuItem1.Text = "Log Out";
            ChildForm.Close();
            if (SystemInfo.designation == "Doctor")
            {
                this.addDeleteToolStripMenuItem.Visible = true;
                this.reportToolStripMenuItem.Visible = true;
                this.prescribeToolStripMenuItem.Visible = true;
                this.storeToolStripMenuItem.Visible = true;
                this.aboutUsToolStripMenuItem.Visible = true;
            }

            else if (SystemInfo.designation == "Receptionist")
            {
                this.patientInformationToolStripMenuItem.Visible = true;
                this.registrationToolStripMenuItem.Visible = true;
                this.reportToolStripMenuItem.Visible = true;
                this.userInformationToolStripMenuItem.Visible = false;
                this.patientInformationToolStripMenuItem.Visible = true;
                //this.tokenPrinToolStripMenuItem.Visible = true;
                this.tokenToolStripMenuItem.Visible = true;
            }

        }
    else
        MessageBox.Show("You have entered invalid information Please check
it!");

        //pnlHome.Hide();
    }

    private void label1_Click(object sender, EventArgs e)
    {

    }

    private void frmMain_Load(object sender, EventArgs e)
    {

    }
}

```

```

private void logOutToolStripMenuItem_Click(object sender, EventArgs e)
{
    logOutToolStripMenuItem.Visible = false;
    logInToolStripMenuItem.Visible = true;
    toolStripMenuItem1.Text = "Log In";
    lblDesg.Visible = false;

    this.addDeleteToolStripMenuItem.Visible = false;
    this.reportToolStripMenuItem.Visible = false;
    this.registrationToolStripMenuItem.Visible = false;
    this.prescribeToolStripMenuItem.Visible = false;
    this.storeToolStripMenuItem.Visible = false;
    this.aboutUsToolStripMenuItem.Visible = false;
}

```

```

private void showAddDeleteToolStripMenuItem_Click(object sender, EventArgs
e)
{
    frmUser ChildForm = new frmUser();
    //ChildForm.MdiParent = this;
    //pnlHome.SendToBack();
    //pxrMain.SendToBack();
    ChildForm.Show();
}

```

```

private void addDeleteToolStripMenuItem_Click_1(object sender, EventArgs e)
{
}

```

```

private void userInformationToolStripMenuItem_Click(object sender, EventArgs
e)
{
    frmReport childform = new frmReport();
    // frmPatientReport childform= new frmPatientReport();
    //childform.MdiParent = this;
    childform.Show();
}

```

```

private void reportToolStripMenuItem_Click(object sender, EventArgs e)
{
}

```

```

e) private void patientsRegistrationToolStripMenuItem_Click(object sender, EventArgs
    {
        frmRegistration childform = new frmRegistration();
        childform.Show();
    }

    private void prescribeToolStripMenuItem1_Click(object sender, EventArgs e)
    {
        frmPrescription childform = new frmPrescription();
        childform.Show();
    }

    private void patientInformationToolStripMenuItem_Click(object sender,
EventArgs e)
    {
        frmPatientReport childform = new frmPatientReport();
        //frmCrystal childform = new frmCrystal();
        childform.Show();
    }

    private void storeToolStripMenuItem_Click(object sender, EventArgs e)
    {
    }

e) private void medicineStoreToolStripMenuItem_Click(object sender, EventArgs
    {
    }

    private void aboutUsToolStripMenuItem1_Click(object sender, EventArgs e)
    {
        MessageBox.Show("We are the group 4 of System Analysis and Design
Course, With the efficient lead of the leader hanjala and other group members(Md.
Gazi Tanvir, Md. Sharifur Rahman, Md. Elias Talukder and Nasir Ahmed), have done
our project. It will be more efficient In our next version!");
    }

    private void label1_Click_1(object sender, EventArgs e)
    {
    }

    private void tokenPrinToolStripMenuItem_Click(object sender, EventArgs e)
    {

```

```

        frmToken childform = new frmToken();
        childform.Show();
    }
}
}

```

5.6.2 Code for the Log In form

```

namespace Hospital_Management_System
{
    public partial class frmLogIn :
    Form
    {
        public frmLogIn()
        {
            InitializeComponent();
        }

        private void frmLogIn_Load(object sender, EventArgs e)
        {

        }

        internal static void show()
        {
            throw new NotImplementedException();
        }

        private void btnClose_Click(object sender, EventArgs e)
        {
            this.Close();
        }

        private void btnLogIn_Click(object sender, EventArgs e)
        {
            bool logInSuccess = false;

            string pass;

            string connStr =
"server=localhost;user=root;database=hms;port=3306;password=root";
            MySqlConnection conn = new MySqlConnection(connStr);
            MySqlCommand cmd = new MySqlCommand();

```

```
cmd.Parameters.Add("@id", MySqlDbType.Int16, 50).Value = txtId.Text;  
cmd.Parameters.Add("@pass", MySqlDbType.String, 50).Value = txtPass.Text;
```

```
try  
{  
    conn.Open();  
  
    string sql = "SELECT u_Name,u_Pass,u_Des from user WHERE  
u_Id=@id";  
    cmd.CommandText = sql;  
    cmd.Connection = conn;  
    cmd.ExecuteNonQuery();  
  
    MySqlDataReader rdr = cmd.ExecuteReader();  
  
    if (rdr.Read())  
    {  
        SystemInfo.name = rdr[0].ToString();  
        pass = rdr[1].ToString();  
        SystemInfo.designation = rdr[2].ToString();  
  
        if (pass == txtPass.Text)  
        {  
            logInSuccess = true;  
        }  
    }  
}  
else  
{  
    MessageBox.Show("Incorrect Id or Password, Please recheck!");  
}  
rdr.Close();  
}  
  
catch (Exception ex)  
{  
    MessageBox.Show(ex.ToString());  
}  
finally  
{
```

```

        if (logInSuccess)
        {
            this.DialogResult = DialogResult.OK;
        }
        else
        {
            //MessageBox.Show("Wrong UserName Or Password");
            this.DialogResult = DialogResult.Cancel;
        }

        conn.Close();
    }
}
}
}

```

5.6.3 Code for the Registration Form

```

namespace Hospital_Management_System
{
    public partial class frmRegistration :
    Form
    {
        public string sid;
        public int id;    public
        frmRegistration()
        {
            InitializeComponent();
        }

        private void lblId_Click(object sender, EventArgs e)
        {
        }

        private void frmRegistration_Load(object sender, EventArgs e)
        {
            string connStr =
            "server=localhost;user=root;database=hms;port=3306;password=root"
            ;
            MySqlConnection conn = new MySqlConnection(connStr);
            try

```

```

        {
            MySqlCommand cmdr = new MySqlCommand();
            conn.Open();
            string sql1 = "SELECT max(p_id) from patient";           cmdr.CommandText =
sql1;           cmdr.Connection = conn;           cmdr.ExecuteNonQuery();
            MySqlDataReader rdr = cmdr.ExecuteReader();           if
(rdr.Read())
            {
                sid = rdr[0].ToString();
                //id = int.Parse(sid);           if
                (int.Parse(sid) == 0)
                {
                    id = 1;           }
                else
                {
                    id = int.Parse(sid) + 1;
                    SystemInfo.pid = id;
                }
            }
            //MessageBox.Show(id.ToString());
        }

        catch (Exception ex)
        {
            MessageBox.Show(ex.ToString());
        }
    finally
    {
        conn.Close();
    }
}

private void cmbMstts_SelectedIndexChanged(object sender, EventArgs e)
{
}

private void btnAdd_Click(object sender, EventArgs e)
{
    string connStr =
"server=localhost;user=root;database=hms;port=3306;password=root";
    MySqlConnection conn = new MySqlConnection(connStr);

    try
    {
        MySqlCo
mmand

```

```

cmdr =
new
MySqlCo
mmand();
    cmdr.Parameters.Add("@id", MySqlDbType.Int16, 50).Value =
SystemInfo.pid;
    cmdr.Parameters.Add("@name", MySqlDbType.String, 50).Value =
txtName.Text;
    cmdr.Parameters.Add("@age", MySqlDbType.Int16, 50).Value =
txtAge.Text;
    cmdr.Parameters.Add("@cell", MySqlDbType.Int24,
20).Value = txtCell.Text;
    cmdr.Parameters.Add("@gen",
MySqlDbType.String, 50).Value = cmbGen.Text;
cmdr.Parameters.Add("@mstts", MySqlDbType.String, 30).Value =
cmbMstts.Text;
    cmdr.Parameters.Add("@adrs",
MySqlDbType.String, 150).Value
=rtbAdrs.Text;
    cmdr.Parameters.Add("@prblm", MySqlDbType.String, 130).Value =
rtbProb.Text;
    cmdr.Parameters.Add("@date",
MySqlDbType.Date).Value = dtpCdate.Value.Date;

    conn.Open();

    string sql = "INSERT INTO patient
(p_Id,p_Name,p_Age,p_Cell,p_Gen,p_Mstts,p_Adrs,p_Prblm,p_Cdate) VALUES
(@id,@name,@age,@cell,@gen,@mstts,@adrs,@prblm,@date)";
    cmdr.CommandText = sql;
cmdr.Connection = conn;
cmdr.ExecuteNonQuery();

}

catch (Exception ex)
{
    MessageBox.Show(ex.ToString());
}

finally
{
    //txtId.Text = "";
txtName.Text = "";
txtAge.Text = "";
cmbGen.Text = "";
cmbMstts.Text = "";
rtbAdrs.Text = "";
rtbProb.Text = "";
txtCell.Text = "";
SystemInfo.pid = SystemInfo.pid + 1;

```

```

        //MessageBox.Show("Data Inserted to DataBase");
    }
}

private void btnClose_Click(object sender, EventArgs e)
{
    this.Close();
}

private void btnClear_Click(object sender, EventArgs e)
{
    //txtId.Text = "";
    txtName.Text = "";
    txtAge.Text = "";
    cmbGen.Text = "";
    cmbMstts.Text = "";
    rtbAdrs.Text = "";
    rtbProb.Text = "";
    txtCell.Text = "";

}
}
}

```

5.6.4 Code for the Prescription Form

```

namespace Hospital_Management_System
{
    public partial class frmPrescription :
    Form
    {
        public frmPrescription()
        {
            InitializeComponent();
        }

        private void frmPrescription_Load(object sender, EventArgs e)
        {
            //txtId.Visible = false;
            //lblId.Text = "Please Click to enter Id for searching";
            string connStr =
            "server=localhost;user=root;database=hms;port=3306;password=root";
            MySqlConnection conn = new MySqlConnection(connStr);

            //Console.WriteLine("Connecting to MySQL...");
        }
    }
}

```

```

        //txtId.Focus();
        try
        {

            MySqlCommand cmd = new MySqlCommand();
            //cmd.Parameters.Add("@id", MySqlDbType.String, 50).Value =
txtId.Text;
            //int id = int.Parse(txtId.Text);
            //string name = txtName.Text;
            conn.Open();

            string sql = "DELETE from prescription";
            cmd.CommandText = sql;          cmd.Connection
= conn;          cmd.ExecuteNonQuery();
            //txtDate.Visible = false;
            //MySqlDataReader rdr = cmd.ExecuteReader();

            //while (rdr.Read())
            // {
            //     txtId.Text = rdr[0].ToString();
            //     txtName.Text = rdr[1].ToString();
            // }

            //rdr.Close();
        }

        catch (Exception ex)
        {
            MessageBox.Show(ex.ToString());
        }
    finally
    {
        conn.Close();
    }
}

private void lblMdcn_Click(object sender, EventArgs e)
{
}

private void checkedListBox1_SelectedIndexChanged(object sender,
EventArgs e)
{
}

private void btnClose_Click(object sender, EventArgs e)
{
}

```

```

        this.Close();
    }

    private void btnAdd_Click(object sender, EventArgs e)
    {
        string connStr =
"server=localhost;user=root;database=hms;port=3306;password=root";
        MySqlConnection conn = new MySqlConnection(connStr);

        try
        {
            MySqlCommand cmdr = new MySqlCommand();
            cmdr.Parameters.Add("@id", MySqlDbType.String, 50).Value = txtId.Text;
            cmdr.Parameters.Add("@name", MySqlDbType.String, 50).Value = txtName.Text;
            cmdr.Parameters.Add("@age", MySqlDbType.String, 50).Value =
txtAge.Text;
            cmdr.Parameters.Add("@cell", MySqlDbType.Int24,
20).Value = txtCell.Text;
            cmdr.Parameters.Add("@gen",
MySqlDbType.String, 50).Value = cmbGen.Text;
            cmdr.Parameters.Add("@mstts", MySqlDbType.String, 30).Value =
cmbMstts.Text;
            cmdr.Parameters.Add("@adrs", MySqlDbType.String, 150).Value =
rtbAdrs.Text;
            cmdr.Parameters.Add("@prblm", MySqlDbType.String, 130).Value =
rtbProb.Text;
            cmdr.Parameters.Add("@date", MySqlDbType.Date).Value =
dtpCdate.Value.Date;
            cmdr.Parameters.Add("@medicine", MySqlDbType.String, 150).Value =
cklstMedicine.Text;
            cmdr.Parameters.Add("@test", MySqlDbType.String, 150).Value =
cklstTest.Text;
            cmdr.Parameters.Add("@prscpn",
MySqlDbType.String, 200).Value = rtbPrscpn.Text;
            cmdr.Parameters.Add("@mdate", MySqlDbType.Date).Value =
dtpMdate.Value.Date;
            conn.Open();
            string sql = "UPDATE patient SET
p_Medicine=@medicine,p_Test=@test,p_Prscpn=@prscpn,p_Mdate=@mdate where
p_Id=@id";
            cmdr.CommandText = sql;
            cmdr.Connection = conn;
            cmdr.ExecuteNonQuery();
        }

        catch (Exception ex)
        {
            MessageBox.Show(ex.ToString());
        }
    }
}

```

```

    }
finally
    {
    }
}

private void btnShow_Click(object sender, EventArgs e)
{
    string connStr =
"server=localhost;user=root;database=hms;port=3306;password=root";
    MySqlConnection conn = new MySqlConnection(connStr);

    //Console.WriteLine("Connecting to MySQL...");
    //txtId.Focus();
    try
    {
        //txtDate.Visible = true;
        MySqlCommand cmd = new MySqlCommand();

        cmd.Parameters.Add("@id", MySqlDbType.String, 50).Value = txtId.Text;

        conn.Open();
        string sql = "SELECT
p_Id,p_Name,p_Age,p_Cell,p_Gen,p_Mstts,p_Adrs,p_Prblm,p_Cdate,p_Medicine,p_
Test,p_Prscpn,p_Mdate from patient WHERE p_Id=@id";
        cmd.CommandText = sql;          cmd.Connection =
conn;
        cmd.ExecuteNonQuery();

        MySqlDataReader rdr = cmd.ExecuteReader();

        if (rdr.Read())
        {
            //dtpJoining.Visible = false;
            txtId.Text = rdr[0].ToString();
            txtName.Text = rdr[1].ToString();
            txtAge.Text = rdr[2].ToString();
            txtCell.Text = rdr[3].ToString();
            cmbGen.Text = rdr[4].ToString();
            cmbMstts.Text = rdr[5].ToString();
            rtbAdrs.Text = rdr[6].ToString();
            rtbProb.Text = rdr[7].ToString();
            dtpCdate.Text = rdr[8].ToString();

            cklstMedicine.Text = rdr[9].ToString();
            cklstTest.Text = rdr[10].ToString();
        }
    }
}

```

```

rtbPrscpn.Text = rdr[11].ToString();
dtpMdate.Text = rdr[12].ToString();
//dtpJoining.Text = rdr[5].ToString();
    //txtDate.Text = rdr[5].ToString();
    //txtSal.Text = rdr[6].ToString();
    //MessageBox.Show("Data found and Showd");
}
else
{
    MessageBox.Show("Data not found in the database");
}

rdr.Close();
}

catch (Exception ex)
{
    MessageBox.Show(ex.ToString());
}
finally
{
    conn.Close();
}

private void btnClear_Click(object sender, EventArgs e)
{
    txtId.Text =
    "";
    txtName.Text =
    "";
    txtAge.Text =
    "";
    cmbGen.Text =
    "";
    cmbMstts.Text =
    "";
    rtbAdrs.Text =
    "";
    rtbProb.Text =
    "";

    txtCell.Text = "";
    //unchecked cklstMedicine.ItemCheck;
    //cklstTest.ItemCheck=unchecked;
    rtbPrscpn.Text = "";
}

private void lblId_Click(object sender, EventArgs e)
{
    //txtId.Visible = true;
    //lblId.Text = "Patient Id";
}

private void txtId_TextChanged(object sender, EventArgs e)
{

```

```

    }

    private void btnReport_Click(object sender, EventArgs e)
    {
        string connStr =
"server=localhost;user=root;database=hms;port=3306;password=root";
        MySqlConnection conn = new MySqlConnection(connStr);

try
    {

        MySqlCommand cmdr = new MySqlCommand();

        cmdr.Parameters.Add("@id", MySqlDbType.String, 50).Value = txtId.Text;
        cmdr.Parameters.Add("@name", MySqlDbType.String, 50).Value = txtName.Text;
        cmdr.Parameters.Add("@age", MySqlDbType.String, 50).Value =
txtAge.Text;
        cmdr.Parameters.Add("@cell", MySqlDbType.Int24,
20).Value = txtCell.Text;
        cmdr.Parameters.Add("@gen",
MySqlDbType.String, 50).Value = cmbGen.Text;
        cmdr.Parameters.Add("@mstts", MySqlDbType.String, 30).Value =
cmbMstts.Text;
        cmdr.Parameters.Add("@adrs",
MySqlDbType.String, 150).Value = rtbAdrs.Text;
        cmdr.Parameters.Add("@prblm", MySqlDbType.String, 130).Value
= rtbProb.Text;
        cmdr.Parameters.Add("@cdate",
MySqlDbType.Date).Value = dtpCdate.Value.Date;
        cmdr.Parameters.Add("@medicine", MySqlDbType.String, 150).Value =
cklstMedicine.Text;
        cmdr.Parameters.Add("@test", MySqlDbType.String, 150).Value =
cklstTest.Text;
        cmdr.Parameters.Add("@prscpn",
MySqlDbType.String, 200).Value = rtbPrscpn.Text;
        cmdr.Parameters.Add("@mdate", MySqlDbType.Date).Value =
dtpMdate.Value.Date;

        conn.Open();

        string sql = "INSERT INTO prescription
(id,name,age,cell,gen,mstts,adrs,prblm,cdate,medicine,test,prscpn,mdate) VALUES (
@id,@name,@age,@cell,@gen,@mstts,@adrs,@prblm,@cdate,@medicine,@test,
@prscpn,@mdate)";
        cmdr.CommandText = sql;
        cmdr.Connection = conn;
        cmdr.ExecuteNonQuery();
        //SystemInfo.tokenid = int.Parse(txtId.Text);

    }

```

```
catch (Exception ex)
{
    MessageBox.Show(ex.ToString());
}

finally
{
    //txtId.Text = "";
    //txtCc.Text = "";
    //txtRoom.Text = "";

    conn.Close();
    //string info = SystemInfo.tokenid.ToString();

    frmPrescriptionPrint report = new frmPrescriptionPrint();
report.Show();
}
}
}
```

Chapter 6

Conclusion

6.1 Conclusion

The project E Prescribing (E-Px) is for computerizing the prescribing system in a hospital. The software takes care of all the requirements of an average hospital prescribing process and is capable to provide easy and effective storage of information related to patients that come to the hospital. The system also provides the facility of backup as per the requirement. The developed project is deployable to

any hospital especially for the huge amount of public hospital. Government has also a plan to deploy this sort of software to ensure more effective care for the patients specially located in the remote area. I have already talked with a government high official. He has appreciated a lot after looking my project.

6.2 Future Scope

E-Prescription System is developed to provide easier and effective way while prescribing in a hospital or other healthcare center. Now this system is capable to add user in the application, patient registration, patient prescription, report generation and sending the report as mail attachment through internet. This sort of option while using this application will ease the prescribing procedure. Besides this, future works related to the present development can be carried out. A few area of the future works are outlined below:

- a) Ensuring more option to prescribe like enriched medicine and test list.
- b) Improving follow-up and patient status updating recording system.
- c) Converting the whole system into web based and include scope for the different pharmacy to know about their medicine market from the application database.

After implementation of the developed system in replace of the conventional paper based hand written prescription system, the patient will get a printed prescription where the specific medicines, tests as well as the dosage are clearly assigned. So the patients will get more reliable service from the hospital. Especially it will save the patient from having wrong medicine for the hazy hand written prescription.

References

[2] http://www.mohfw.gov.bd/index.php?option=com_content&view=frontpage&Itemid=1&lang=en; accessed on 04.03.12

[3] <http://msdn.microsoft.com/en-us/library/ms123401.aspx>; accessed on 14.05.12

[4] <http://www.mysql.com/why-mysql/isv-oem-corner/security/>; accessed on 19.05.12

- [5] <http://www.csharp-station.com/Tutorial.aspx>; accessed on 12.03.12
- [6] <http://www.java2s.com/Tutorial/CSharp/CatalogCSharp.htm>; accessed on 24.06.12
- [7] ipedia.org/wiki/Electronic_prescribing; accessed on 17.07.12
- [8] Joel Murach, murach's C# 2008, Shroff Publishers and Distributors Pvt. Ltd, 1st Edition, May 2008
- [9] Elias M. Awad, System Analysis and Design, Galgotia Publications (P) Ltd. 2nd Edition, 2003.