

MINISTRY OF SCIENCE AND HIGHER EDUCATION OF THE RUSSIAN FEDERATION

Federal State Autonomous Educational Institution of High Education
South Ural State University (National Research University)
School of Electrical Engineering and Computer Science
Department of Computer Science

THESIS IS CHECKED

Reviewer,
Cand. Sci., Assoc. Prof. of IITTMCS
SUSHU

_____ O.A. Dmitrieva

“ ___ ” _____ 2019

ACCEPTED FOR THE DEFENSE

Head of the department, Dr. Sci., Prof.

_____ L.B. Sokolinsky

“ ___ ” _____ 2019

**DEVELOPMENT OF A WEB APPLICATION
"ELECTRONIC MEDICAL CARD"**

GRADUATE QUALIFICATION WORK
SUSU-02.04.02.2019.308-643.GQW

Supervisors,
Cand. Sci., Assoc. Prof.

_____ S.A. Ivanov

Senior Lecturer

_____ N.S. Silkina

Author,
the student of the group CE-229

_____ A.A.S. Sultan

Normative control

_____ O.N. Ivanova

“ ___ ” _____ 2019

MINISTRY OF SCIENCE AND HIGHER EDUCATION OF THE RUSSIAN FEDERATION

Federal State Autonomous Educational Institution of High Education
South Ural State University (National Research University)
School of Electrical Engineering and Computer Science
Department of Computer Science

APPROVED

Head of the department,
Dr. Sci., Prof.

_____ L.B. Sokolinsky

“ ” _____ 2019

TASK

of the master graduate qualification work

For the student of the group CE-219

Sultan Abdulrahman Ahmed Sultan

In master direction 02.04.02

“Fundamental Informatics and Information Technologies”
(master program “Database Technologies”)

1. The topic (approved by the order of the rector from 25.04.2019 No. 899)

Development of a web application "Electronic medical card".

2. The deadline for the completion of the work: 01.06.2019.

3. The source data for the work

3.1. Freeman A. Pro Entity Framework Core 2 for ASP.NET Core MVC.
Berkeley, CA: Apress, 2018. 650p.

3.2. Mistry R., Misner S. Introducing Microsoft SQL Server 2014 Technical
Overview. Redmond, W: Microsoft Press, 2014. 128 p.

4. The list of the development issues

4.1. To define the problem statement;

4.2. To make a comparative analysis of analogues;

4.3. To choose development tools;

4.4. To determine functional and non-functional requirements;

4.5. To design the database;

4.6.To design and to implement the web-application;

4.7.To test the system.

5. Issuance date of the task: 09.02.2019.

Supervisor

Senior Lecturer

N.S. Silkina

The task is taken to perform

A.A.S. Sultan

TABLE OF CONTENTS

INTRODUCTION.....	5
1. THE ANALYSIS OF THE SUBJECT AREA	7
2. DESIGNING OF SYSTEM	14
3. IMPLEMENTATION OF WEB-APPLICATION	22
4. TESTING OF WEB-APPLICATION	32
CONCLUSION	35
REFERENCES.....	36

INTRODUCTION

Managing health information using information technology is an important part of the changing health-care system. A health-care information management system is defined as software involving of a collection for the procedures and the programs with the requirements of all necessary actions such as entering, storing, retrieving, updating and manipulating data having suitable capacity to keep the integrity, security and confidentiality with achieving management, legal and accounting requirements. Proper implementation of information technology makes it easy for healthcare providers to store, share and access the health information. With an increase in access to larger computers and new advances in information technology resulted the development of more efficient data management system. However, all these development are within the network of a particular hospital or within the network of a limited number of hospitals. The patient cannot utilize their medical data outside the network as patient wishes. The invention of smart card concept in 1970s and the development of the internet in the 1980s envisaged new fields of health care information management system using smart health card [1]. In hospitals, there is a need for an electronic system for the health card for people, where the card contains their personal information and information to review the doctors and the hospital and the latest cases and symptoms for each person where the system provides the effort in analyzing the causes of some of the diseases experienced by the person through reading the electronic book via the national number

To the person and know all the symptoms of the recent exposure to the system also maintains the confidentiality of information, as the system helps the doctor in his work instead of the work of comprehensive tests may be taken at a later date and recorded in the electronic health card, which helps easy to detect cases of emergency in people if the card keeps a record Complete for all doctors and hospital reviews via the website if doctors need to speed up instead of traditional ways of discovering diseases or searching through traditional health

cards. So, developing a web-application –"Electronic medical card" to carry the entire medical records of a patient is an actual problem. This system will allow you to store and update medical records of patients, including the personal profile of the patient, data on current and past treatment, as well as diagnostic data.

1. THE ANALYSIS OF THE SUBJECT AREA

Problem statement

The main goal of this project is a developing an electronic medical card to carry the entire medical records of a patient. To succeed this objective it is essential to solve the following tasks:

- 1) designing the database;
- 2) designing the system;
- 3) developing system;
- 4) testing system.

Comparative analysis of analogues

There are two approaches to solve the problem of storing patient's medical data:

- using physical device – smart card;
- Medical database stored on Internet cloud.

Consider the first approach. A smart card is a physical card that has an embedded integrated chip that acts as a security token. Smart cards are typically the same size as a driver's license or credit card and can be made out of metal or plastic. They connect to a reader either by direct physical contact (also known as chip and dip) or through a short-range wireless connectivity standard such as *radio-frequency identification* (RFID) or *near-field communication* (NFC).

The chip on a smart card can be either a microcontroller or an embedded memory chip. Smart cards are designed to be tamper-resistant and use encryption to provide protection for in-memory information. Those cards with microcontroller chips can perform on-card processing functions and can manipulate information in the chip's memory.

This approach is used in many areas. For example, payment cards, smart cards to control access to physical locations and so on. One of the important uses of a smart card is to securely store patient medical records [2, 3].

Consider the second approach. Cloud data storage solutions cater to more advanced infrastructure technology because they give users a more flexible way to access data. Several different types of cloud deployments are available to healthcare organizations.

However, IT decision-makers need to decide how much control over their data these deployments require, which often leads to hybrid cloud solutions.

Healthcare organizations have many kinds of data, from *Electronic Health Records* (EHRs) to unstructured patient data, some of which needs to be accessed quickly and frequently. Other types of data will be held in a repository until it's needed.

Entities don't need to choose between on premise or cloud storage deployments. Instead, healthcare organizations can choose what data is stored using a variety of methods that suit each unique infrastructure.

The Patients go to new doctors they need to fill out paper medical-history forms as an application form. What's more, over time, records can become spotty, incomplete, and difficult to access. This leads to both inefficiencies in the medical-record system, which cost money, and medical faults, which can cost lives.

The researchers and the entrepreneurs hope to change that by giving each patient a smart card holding his or her whole medical history as data stored in database. This approach may verify difficult to implement in the U.S., because of security fears and compatibility issues, but the technology has the potential to convert health care in countries that have unified health systems, or where there's inadequate infrastructure for sharing records in other ways [4].

Researchers in the U.K. have developed the My Care card, which is roughly the size and shape of a credit card, with a fold-out USB plug [5]. Another project, Smart Care, first implemented in Zambia, has lately expanded to Ethiopia and South Africa and demonstrates the potential for card-based systems in parts of the world with incomplete infrastructure [6].

The My Care card was developed at City University London; the software for it was developed at Coventry University. The fact that anybody can download

and view the code, and anyone can contribute to efforts to improve or expand on it, confers rewards.

Developers could make new software that interfaces with the data on the card—for instance, to automatically recognize incompatible prescriptions and show a warning to pharmacists. Open-source software may also offer increased security, because the software can be scrutinized openly for serious flaws. Security is a main matter surrounding medical ID cards, which store potentially sensitive private information. People fear the possibility of losing their entire medical history as simply as they might lose a wallet.

The My Care card is also meant to interface as simply as possible across a variety of computers and operating systems. Rather than requiring installation on a computer, the card's software runs directly from the card itself.

At the current stage of development, PINs and some degree of encryption protect the data on the card. Pacino's Kyriacos, head of the project at City University London, says that more secure encryption will be implemented further along in the development process [7].

The encryption gives patients and doctors different levels of access. Patients can update personal information, such as next of kin or contact information, but the software allows only professionals such as doctors to edit prescriptions.

No matter how flexible the software on the card becomes, however, it will not be able to automatically work with every hospital database—health-care providers will still need to cooperate to ensure compatibility.

In the U.K., where government-paid health care is the norm, medical-record systems are already more nearly standardized than they are in the U.S., so it may be easier to establish a universal electronic record system.

The card-based system is “not a good fit” for the U.S., says John Halakha, chairman of the Healthcare Information Technology Standards Panel and CIO (chief information officer) of both Harvard Medical School and Beth Israel Deaconess Medical Center, in Boston. “We tend to be more network- and mobile-

centric,” he says, and “carrying around a card, which is common in Europe, is not our culture.”

He is more optimistic about the chief alternative to card-based electronic medical records: the cloud. The U.S. needs “Web-based personal health records in the cloud, available anywhere at any time without a card,” he says. Such a network-based alternative may require greater changes to existing health-care networks, but it is likely to give physicians even better access to important patient information.

Efforts to achieve an electronic medical-record system in the U.S., whether card- or cloud-based, are complicated by the fact that hospitals essentially compete with one another for patients. This means they have a vested interest in keeping patients—and their records—within their networks and their networks alone.

So far, the U.K. has spent £2.7 billion on a National Health Service program for IT, according to a May 18 government report. Meanwhile, in the U.S., Congress included huge cash incentives for health IT in the 2009 stimulus package, offering millions to hospitals that demonstrate “meaningful use” of electronic records.

Smart-card systems could be well suited to low-tech environments such as rural Zambia. Zambia lacks the infrastructure for a network-based system, but spotty telecommunications and unreliable power supplies can’t compromise information stored on cards. The Smart Care project was initially developed in cooperation with the Zambian government, but it has expanded its scope, with Smart Care cards now deployed in Zambia, Ethiopia, and South Africa.

As for the My Care card, Kyriacos has high hopes for the “universal” concepts that underpin it. The card could, he says, meet the “need to have control and ownership over our medical information.”

The Open Clinical Web site [<http://www.openclinical.org/home.html>] is targeted in particular at healthcare the professionals and the managers, medical information and computer scientists. It is designed to be a "one-stop shop" for anyone interested in learning about and following developments on advanced knowledge management technologies for healthcare such as point-of-care decision support systems, "intelligent" guidelines and clinical workflow.

Medical chain [<https://medicalchain.com/en/>] uses block chain technology to securely store health records and maintain a single version of the truth. The different organizations such as doctors, hospitals, laboratories, pharmacists and health insurers can request permission to access a patient's record to serve their purpose and record transactions on the distributed ledger. Medical chain provides solutions to today's health record problems. The platform is built to securely store and share electronic health records. By digitizing health records and empowering users we can leverage countless industry synergies.

American Cancer Society [<https://www.cancer.org/>]: At the American Cancer Society has the mission to free the world from cancer, where used for funding and conducting research, sharing expert information, supporting patients, and spreading the word about prevention.

In this work, the electronic medical card has been created to reach the requirements of the health organizations by saving and sorting patient information and monitoring the health status. In this system, the medical staff can describe the appropriate treatment for the patient by observing the previous reports including recipes and Diagnosis stored in the system. The main objective of the electronic medical card is to help the doctor in making the appropriate decision for the patient as well as reduce the cost of unnecessary treatment for the patients and health organizations.

Choice of development tools

For developing web-application I chose Microsoft ASP.net and SQL Server 2014.

ASP.NET is a server-side web application framework designed for web development to produce dynamic web pages. Microsoft was developed the ASP.NET as .NET Framework environment to allow programmers build dynamic web sites, web applications and web services [8, 9].

It was first released in January 2002 with version 1.0 of the .NET Framework, and is the successor to Microsoft's Active Server Pages (ASP)

technology. ASP.NET is built on the Common Language Runtime (CLR), letting the programmers to write ASP.NET code using any supported .NET language such as c# or visual basic. The ASP.NET SOAP extension framework allows ASP.NET components to process SOAP messages.

ASP.NET's successor is ASP.NET Core. It is a re-implementation of ASP.NET as a modular web framework, together with other frameworks like Entity Framework. The new framework uses the new open-source .NET Compiler Platform (codename "Roslyn") and is cross platform. ASP.NET MVC, ASP.NET Web API, and ASP.NET Web Pages (a platform using only Razor pages) have merged into a unified MVC 6[10].

Advantages of ASP.NET

1. It drastically reduces the amount of code required to build large applications.
2. With built-in Windows authentication and per-application configuration, your applications are safe and secured.
3. It provides better performance by taking advantage of early binding, just-in-time compilation, native optimization, and caching services right out of the box.
4. The framework is perfected by a rich toolbox and designer in the Visual Studio integrated development environment. WYSIWYG editing, drag-and-drop server controls, and automatic deployment are just a few of the features this powerful tool delivers.
5. The source code and HTML are together therefore its pages are easy to maintain and write. Also the source code is executed on the server side and the user just see the result. This provides a lot of power and flexibility to the web pages.

Advantages of ASP.NET compared to PHP technology:

- ASP.NET provides the freedom to choose from multiple languages such as C#, Visual Basic.NET, C++, etc.;
- the SQL security feature that ASP.NET provides is automatically applied to the applications and has unlimited data storage;

- with ASP.NET you get multiple .NET libraries which is not the case in PHP development;
- ASP.NET allows threading of codes that allows multiple codes to run at the same time;
- ASP.NET developers are paid much better than any PHP developer because of there are less ASP.NET developers in the market.

Microsoft SQL Server is a relational database management system established by Microsoft. As a database server, it is a software product with the primary function of storing and retrieving data as requested by other software applications—which may run either on the same computer or on another computer across a network (including the Internet) [11].

Microsoft markets at least a dozen different editions of Microsoft SQL Server, aimed at different audiences and for workloads ranging from small single-machine applications to large Internet-facing applications with many concurrent users [12].

It can be concluded that for development of the Electronic medical card both technologies ASP.Net and PHP are suitable. Both technologies are also relatively similar in programming paradigm, for instance, both supports for object oriented programming [13]. However, there are certain architectural differences including the way script is compilation and events are handled. While comparing the popularity, PHP is found to be more popular than ASP.Net. Both technologies have rich APIs for handling various database connections. Nonetheless, it can be established that ASP.Net is reasonably compatible with MS SQL Server whereas PHP with MySQL. Moreover, for generating dynamic content, particularly from database, ASP.Net provides various automated tools for handling connection, displaying content in tabular form, and also supports automatic paging, as a result, making the development more rapid. On the contrary, for implementation of these functionalities, in PHP, it depends on logic derived by developer and by taking use of some in-built functions, making it often time consuming [14].

2. DESIGNING OF SYSTEM

Functional and non-functional requirements

The problem statement defined the following functional requirements.

The system must do the following.

1. There are two types of users: Doctor and Patient. Each of them can authorization with login information data (username and password).
2. Doctor can see information about his patients and their diagnoses.
3. Patient can register in the system.
4. Patient can see list of doctors and choose the date of review.

Use case diagram

UML is a language for modeling different aspects of system design [15, 16, 17]. There is a use case diagram, an activity diagram, a state chart diagram, a sequence diagram, and collaboration diagram.

The first step for system design is a creating use case diagram. Use case diagram tell us about functions of our system.

Use case diagram consists of a set of use cases and actors and their relationships.

The use case diagram of the system “Electronic medical card” consists of two actors: *Doctor* and *Patient*.

Each of them can authorization with login and password.

It is not shown in use case diagram.

Doctor can see information about his patients and their diagnoses.

Doctor can also create Recipe.

Patient can register in the system and register in the electronic medical card.

Later Patient can see list of doctors and choose the date of review.

Use case diagram of “Electronic medical card” system is shown on fig. 1.

Three use cases are available for Patient.

Four use cases are available for Doctor.

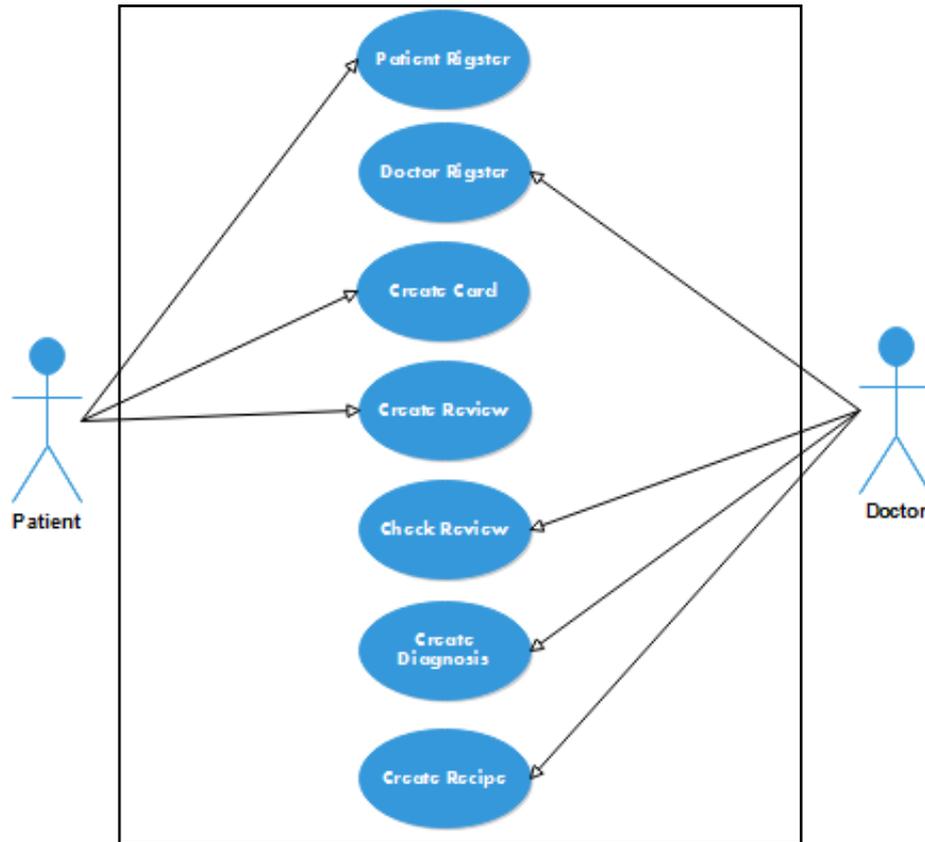


Fig. 1. Usecase diagram

Database description

A database is a data structure that stores organized information.

Most databases contain multiple tables, which may each include several different fields.

Simply it's a collection of organized information, usually as a set of related lists of similar entries. The data is often organized so that it is easily accessible.

In this system I used DBMS MS SQL SERVER [18] because it's very fast, reliable, and easy to use, free to download and students can use it, it uses standard SQL.

Database schema of my project is shown on fig. 2.

It consists of 7 tables.

In the following we list all tables in this database with some of data.

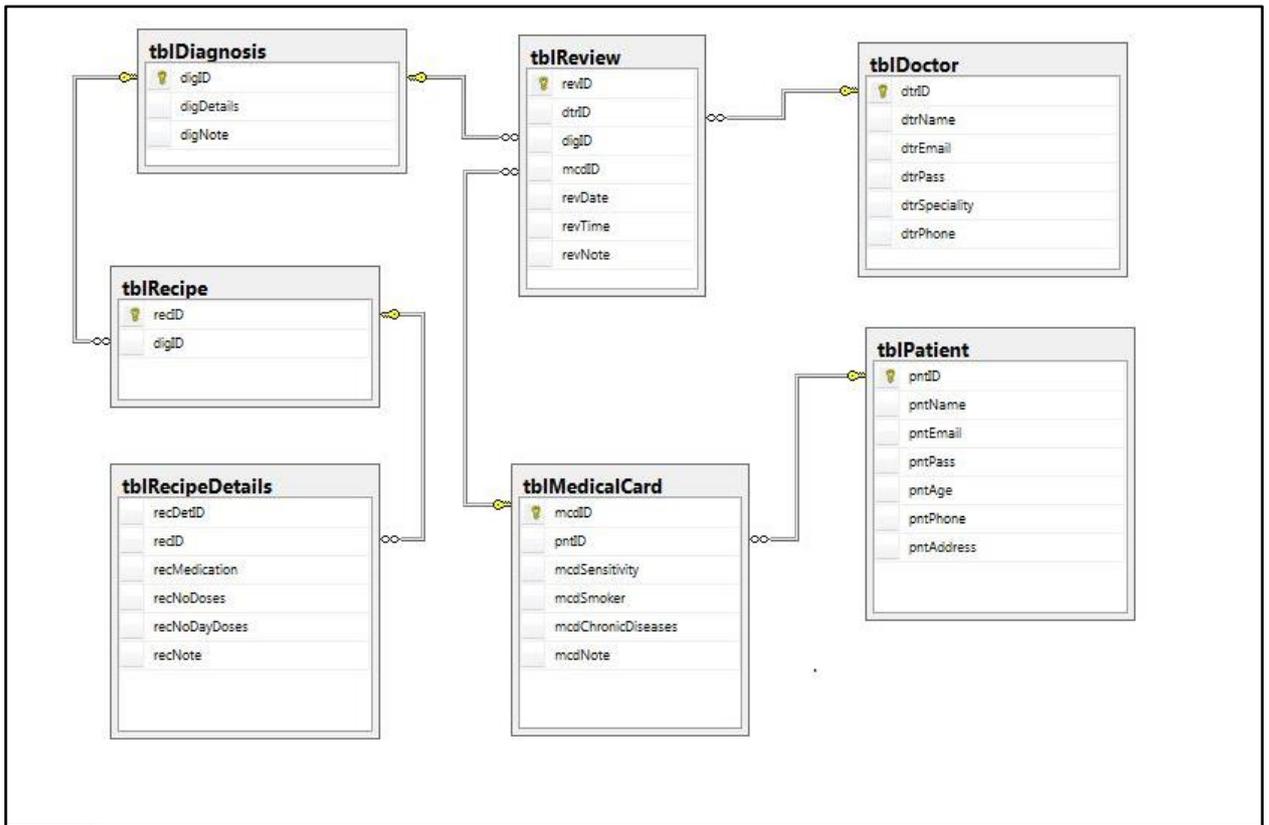


Fig. 2. Database schema

Table “tblDiagnosis”

This table has three columns. Column digID as a primary key with int datatype. The revID as text datatypes with allow null property. The column digDetails stores information about diagnosis and is shown on fig. 3.

digID	revID	digDetails
1	3	sugar
2	4	headache
3	5	flu

Fig. 3. tblDiagnosis

Table “tblDoctor”

This table has seven columns. dtrID as primary key with int datatype. The dtrName, dtrPass and dtrSepciality are nvarchar(50) datatype.

The dtrEmail is varchar(50) datatype with allow null property and dtrPhone is int datatype with allow null property as shown on fig. 4.

dtrID	dtrGUID	dtrName	dtrEmail	dtrPass	dtrSpeciality	dtrPhone
1	255eb9a3-789e-412d-986b-9...	doctor1	doc1@doc1.com	123	Arthritis diseases	123
2	459eb9a3-789e-412d-986b-9...	doctor2	doc2@doc2.com	321456	Blood diseases	96455172
3	996eb9a3-789e-412d-986b-9...	doctor3	doc3@doc3.com	456321	Ear, Nose and Throat	96400215

Fig. 4. tbIDocror

Table “tblMedicalCard”

This table has five columns. pntID as primary key with int datatype.

The mcdSensitivity, mcdSmoker and mcdChronicDiseases are a bit datatype (default value is False).

The mcdNote is text datatype with allow null property as shown in following fig. 5.

pntID	mcdSensitivity	mcdSmoker	mcdChronicDi...	mcdNote
1	True	True	True	
2	True	True	True	NULL
3	False	False	False	NULL
4	True	False	True	NULL

Fig. 5. tblMedicalCard

Table “tblPatient”

This table has seven columns pntID as primary key with int datatype.

The pntName and pntPass are a nvarchar(50) datatype.

The pntEmail is varchar (50) datatype with allow null property.

The pntAge and pntPhone are int datatype with allow null property.

The pntAddress is nvarchar(150) datatype with allow null property as shown in following fig. 6.

pntID	pntName	pntEmail	pntPass	pntAqe	pntPhone	pntAddress
1	ali	ali@ali.com	123	25	123	baghdad
2	ahmed	ahmed@ahme...	1234	30	964123456	Basrah
3	rami	rami@rami.com	123123	23	96425135	Dyala
4	mohammed	mohammed@...	m123m	35	96400145	Erbil

Fig. 6. tbIPatient

Table “tblRecipe”

This table has two columns recID as primary key with int datatype. The digID is int datatype with allow null property as shown in following (fig. 7).

recID	digID
1	1
2	2
3	3

Fig. 7. tblRecipe

Table “tblRecipeDetails”

This table has six columns recDetID as primary key with int datatype.

The recID is int datatype with allow null property.

The recMedication is a nvarchar (50) datatype with allow null property.

The recNoDoses and recNoDayDoses are int datatype with allow null property.

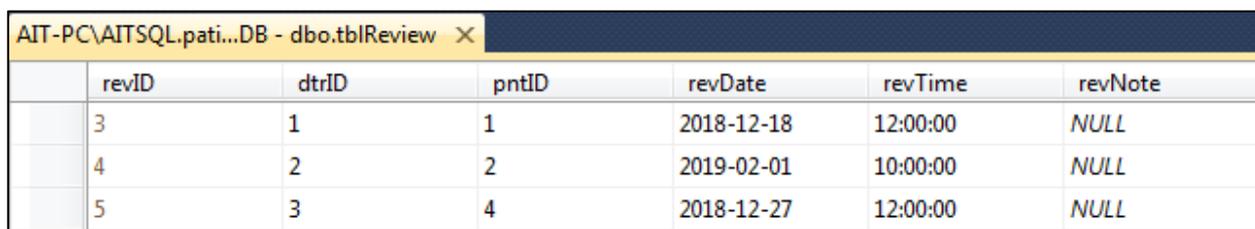
The recNote is text datatype with allow null property as shown in following fig. 8.

recDetID	recID	recMedication	recNoDoses	recNoDayDoses	recNote
1	1	glucophage	1	3	NULL
2	1	relief	2	6	NULL
3	2	paracetamol	3	7	NULL

Fig. 8. tblRecipeDetails

Table “tblReview”

This table has six columns. revID as primary key with int datatype. The dtrID and pntID are int datatype with allow null property. The revDate is date datatype with allow null property. The revTime is time (7) datatype with allow null property. The revNote is text datatype with allow null property as shown in following fig. 9.



revID	dtrID	pntID	revDate	revTime	revNote
3	1	1	2018-12-18	12:00:00	NULL
4	2	2	2019-02-01	10:00:00	NULL
5	3	4	2018-12-27	12:00:00	NULL

Fig. 9. tblReview

Development of user interfaces

The user interface is the aggregate of means by which the user has interacted with the system device such as hardware and computer program or other complex a particular machine and [19]. The part of an interactive computer program sends messages to and receives instructions from a terminal user. User Interface Design and Ergonomics deals with analysis, design, implementation and evaluation of user interface design. I have designed the user interface of the developing system, which is shown in following fig.10.

Figure 10 represents the project flowchart that illustrates the main components of the pages created. For example, the doctor can login to his personal page only after inserting his username and password in the sign in page. After that the doctor has two options, he can see and edit his personal profile information or he can insert directly to the review page so he can review the available medical card, information of diagnoses and recipe. The same thing for the patient, where he can login to his personal page after inserting his username and password in the sign in page. After that he can simply see his reviews, information of diagnoses and recipe that made by the doctor.

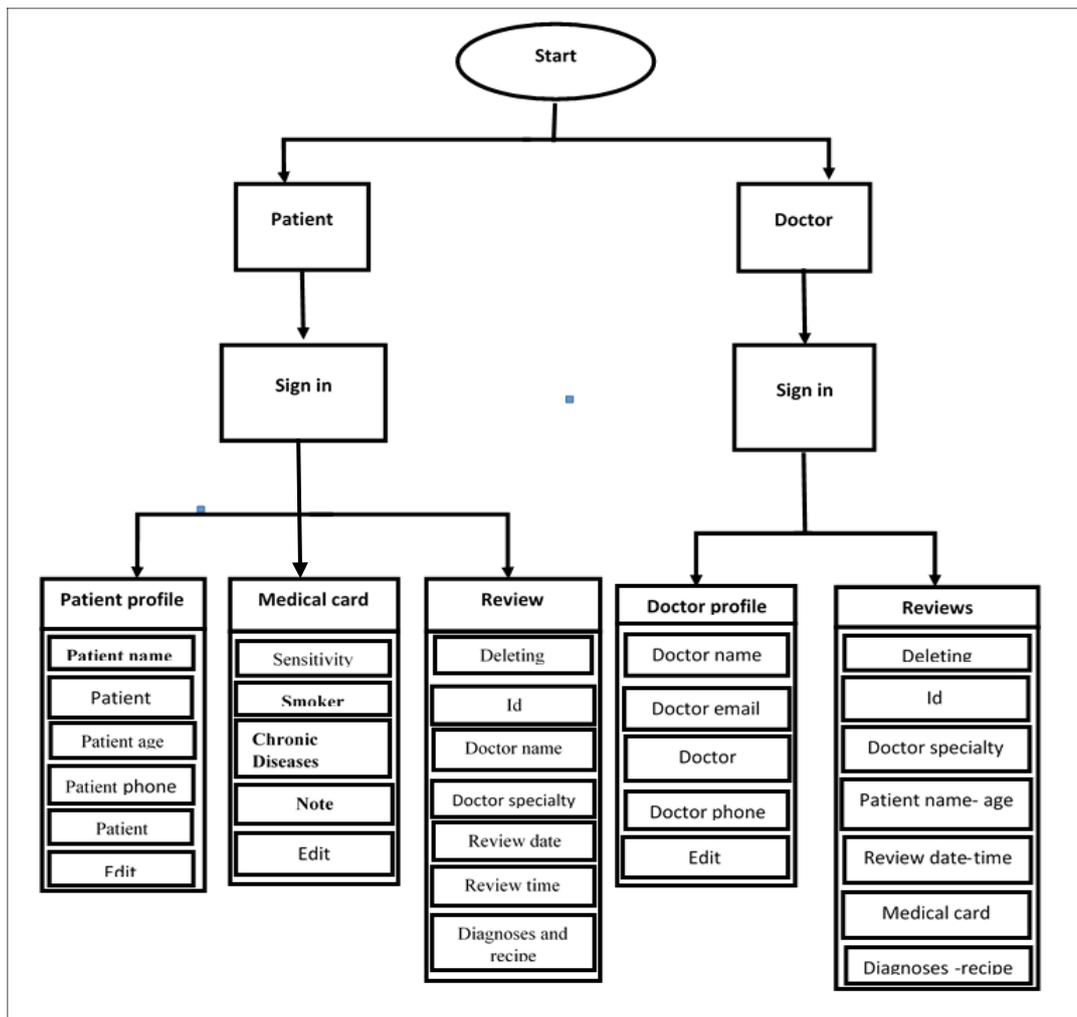


Fig. 10. Website structure

This system architecture has many components. These components work together to server patients and doctors. There is start point when user choose sign in form and type the username and password. The system has to display user interface (patient or doctor). The doctor after login can check all reports for patients and make decision by using all historical information about patient. Each doctor can edit his information and change the name, password, etc. The patient also has login page after login he can edit his information. The main purpose for the patient's pages are booking the date with doctors and review all the historical visitation and information about diagnostics and recipes.

There is a card for patients, this card has ID for each patient. The doctors can use this card to browse all patient information. By using this card the doctors can

easily register the new diagnostics and recipes. Additionally the doctors can save all note that related to this patient such us new visiting date and the future diagnostics and so on. Finally the system save all records that registered by doctors and connect them with ID card.

3. IMPLEMENTATION OF WEB-APPLICATION

Components of the project

This system has been programmed by using ASP.NET pages as mentions above and we have added many libraries that we used to make connection between pages and database. Figure 11 below illustrates all pages created for this project. Folder “css” consists files for the nice look for pages. Additionally, there is Web.Config that we used to specify the setting for every ASP.NET application on web server. The Web.Config is standard and human-read-able XML files that we can open and modify with any text editor. The database also added in same directory to make connection and fetch the records faster.

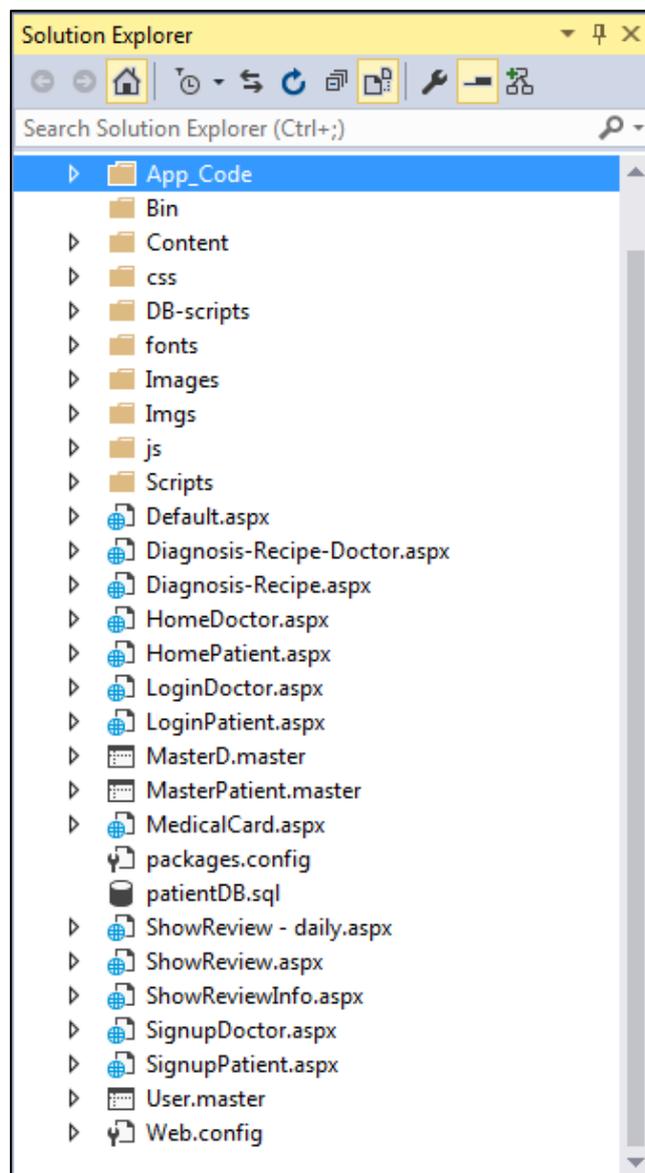


Fig. 11. Project components (folders, pages, database, and configuration files)

There is a page with .master extension with a predefined layout that can include static text, HTML elements and server controls. This page offers a template for one or more web forms. When users request the content page, ASP.NET merges the layout of the master page with the content of the content page and produce output. A file with the ASPX file extension is an Active Server Page Extended file that's designed for Microsoft's ASP.NET framework. They're also called .NET Web forms. ASPX files are generated by a web server and contain scripts and source codes that help communicate to a browser how a web page should be opened and displayed.

Examples of codes

Fig. 12 shows sign in page code. This a complete sign in code for our system.

We have created a SQL connection by using (SqlConnection) object with necessary data such as select text and dataset object. The Click event has been added and used as the click event handler for a button (Button1) to run login code.

The code in figure 13 shows register page code.

```
1 using System;
2 using System.Collections.Generic;
3 using System.Linq;
4 using System.Web;
5 using System.Web.UI;
6 using System.Web.UI.WebControls;
7 using System.Data.SqlClient;
8 using System.Configuration;
9
10 public partial class Home : System.Web.UI.Page
11 {
12     protected void Page_Load(object sender, EventArgs e)
13     {
14         if (!IsPostBack)
15         {
16             LblMsg.Text = "";
17         }
18     }
19
20     protected void Button1_Click(object sender, EventArgs e)
21     {
22         SqlConnection conn = new SqlConnection(ConfigurationManager.ConnectionStrings["patientDBConnectionString"].ConnectionString);
23         conn.Open();
24         SqlCommand com = new SqlCommand("select * from tblDoctor where dtrEmail=@email and dtrPass=@pass", conn);
25         com.Parameters.AddWithValue("@email", TxtDtrEmail.Text.ToLower());
26         com.Parameters.AddWithValue("@pass", TxtDtrPass.Text);
27         SqlDataReader dr = com.ExecuteReader();
28         if (dr.Read())
29         {
30             Response.Cookies["dID"].Value = dr["dtrID"].ToString();
31             Response.Cookies["dID"].Expires = DateTime.Now.AddDays(1);
32             Response.Cookies["dName"].Value = dr["dtrName"].ToString();
33             Response.Cookies["dName"].Expires = DateTime.Now.AddDays(1);
34             Response.Redirect("~/HomeDoctor.aspx");
35         }
36         else
37         {
38             LblMsg.Text = "E-mail or password Incorrect";
39         }
40         conn.Close();
41     }
42 }
```

Fig. 12. Sign in page code

We also used the click event handler to run this code.

This code used parameter (InserParameters) to insert the data in database this technique is more secure because it protect page from the SQL-injection attacks.

The example for displaying the data from database is shown in figure 14.

We also used SqlConnection object.

After run the select sql-query fetch records should converted to dataset such as SqlDataReader object then display them in page.

The dataset can display in many objects from ASP.NET such as Gridview.

```
3 using System.Linq;
4 using System.Web;
5 using System.Web.UI;
6 using System.Web.UI.WebControls;
7 using System.Data.SqlClient;
8 using System.Configuration;
9 using System.Drawing;
10 namespace
11 {
12     public partial class Home : System.Web.UI.Page
13     {
14         protected void Page_Load(object sender, EventArgs e)
15         {
16         }
17
18         protected void Button1_Click(object sender, EventArgs e)
19         {
20             SqlConnection conn = new SqlConnection(ConfigurationManager.ConnectionStrings["patientDBConnectionString"].ConnectionString);
21             conn.Open();
22             string checkUser = "select count(*) from tblPatient where pntEmail=@email";
23             SqlCommand cmd = new SqlCommand(checkUser, conn);
24             cmd.Parameters.AddWithValue("@email", TxtPntEmail.Text.ToLower());
25             int temp = Convert.ToInt32(cmd.ExecuteScalar().ToString());
26             if (temp == 1)
27             {
28                 lblMsg.Text = "User already Exists";
29                 TxtPntEmail.Focus();
30             }
31             else
32             {
33                 SqlDSignupPnt.InsertParameters["pntName"].DefaultValue = TxtPntName.Text.ToLower();
34                 SqlDSignupPnt.InsertParameters["pntEmail"].DefaultValue = TxtPntEmail.Text.ToLower();
35                 TxtPntName.Text = "";
36                 TxtPntEmail.Text = "";
37                 TxtPntPass.Text = "";
38                 TxtPntAge.Text = "";
39                 TxtPntPhone.Text = "";
40                 TxtPntAddress.Text = "";
41                 lblMsg.ForeColor = Color.Green;
42                 lblMsg.Text = "inserted!";
43                 Response.Redirect("~/LoginPatient.aspx");
44             }
45             conn.Close();
46         }
47     }
48 }
```

Fig. 13. Register page code

```
protected void BtnCreate_Click(object sender, EventArgs e)
{
    int rowCount = 0;
    SqlConnection con = new SqlConnection(ConfigurationManager.ConnectionStrings["patientDBConnectionString"].ConnectionString);
    SqlCommand cmd = new SqlCommand("select * from tblReview order by revDate desc, revTime desc", con);
    con.Open();
    SqlDataReader lastDTRdr = cmd.ExecuteReader();
    if (lastDTRdr.Read())
    {
        getLastDate = Convert.ToDateTime(lastDTRdr["revDate"]);
        getLastTime = Convert.ToDateTime(lastDTRdr["revTime"].ToString());
        chkDateTime = true;
    }
    else
    {
        chkDateTime = false;
    }
    con.Close();
    if (chkDateTime == true)
    {
        SqlCommand dateCountCMD = new SqlCommand("select count(*) from tblReview where revdate=@revdate", con);
        dateCountCMD.Parameters.AddWithValue("@revdate", getLastDate);
        con.Open();
        rowCount = (int)(dateCountCMD.ExecuteScalar());
        con.Close();
        if (getLastDate >= DateTime.Now.Date)
        {
            if (rowCount < 10)
            {
                if (rowCount > 0)
                {
                    reviewDate = getLastDate;
                    reviewTime = getLastTime.AddMinutes(30);
                }
                else
                {
                    reviewDate = DateTime.Now.Date;
                    reviewTime = #Time;
                }
            }
            else
            {
                reviewDate = getLastDate.AddDays(1);
                reviewTime = #Time;
            }
        }
    }
}
```

Fig. 14. Review page code

Interfaces of the application

During the implementation of the pre-degree practice, I have implemented several functions of the web application. For the implementation of system I used ASP.Net technology. I implemented user authorization and partially implemented the Patient interface and Doctor interface.

Any guest can browse the main page, this main page holds several links or buttons such as ("home", "sign in", "Contact Us", "information about") as in the fig. 15.

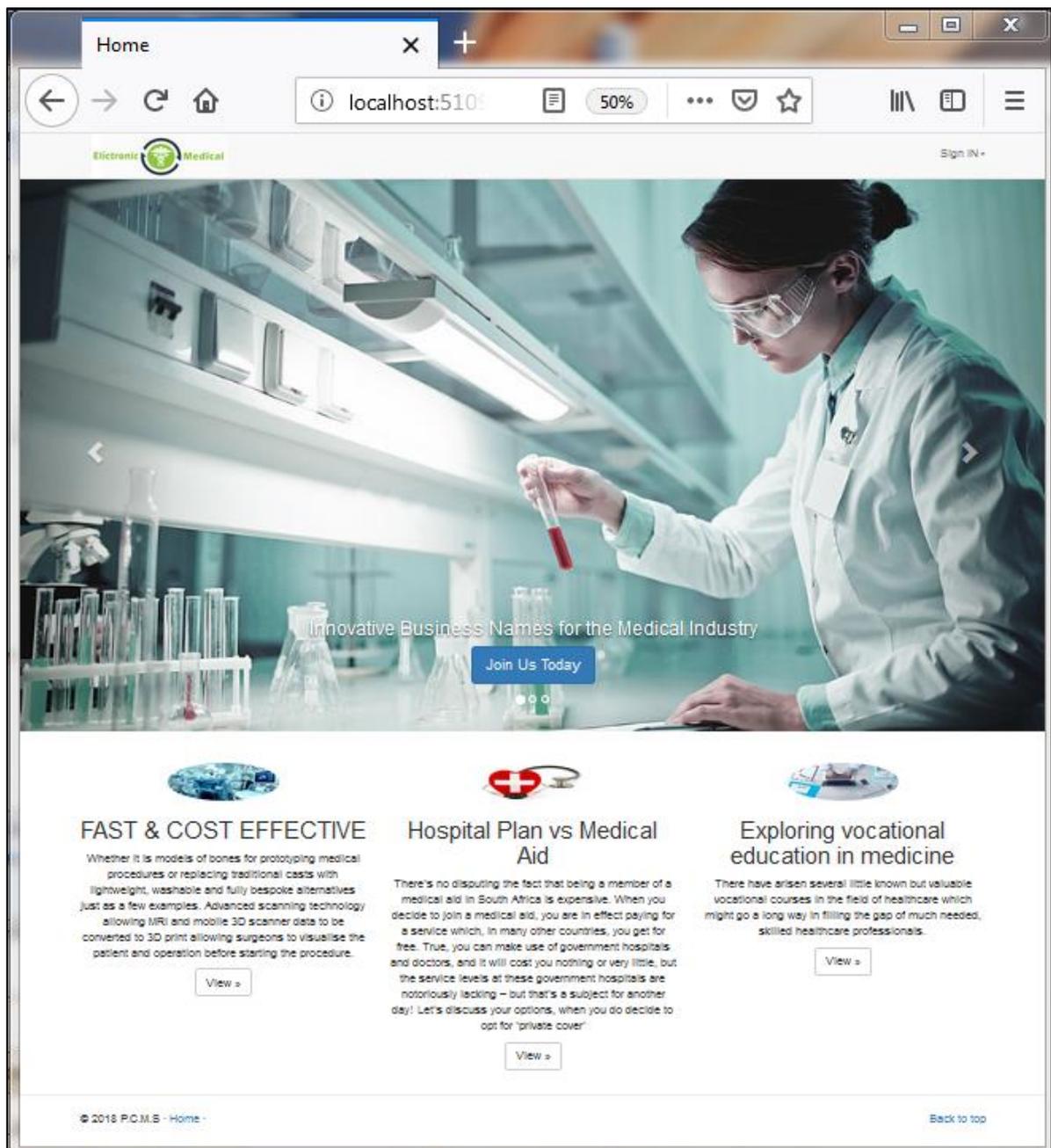


Fig. 15. The main page

The doctor that visits the main page can create new account by insert his information ("Name", "Email", "password", "Specialty", "phone"). He can do that by select "Create an account" in the sign in form as shown in fig. 16.

The screenshot shows a web browser window titled "Electronic Medical Card" with the URL "localhost:5105". The page header features the "Electronic Medical" logo and a "Home" link. The main content area is titled "Registration Doctor Account" and contains the following form fields:

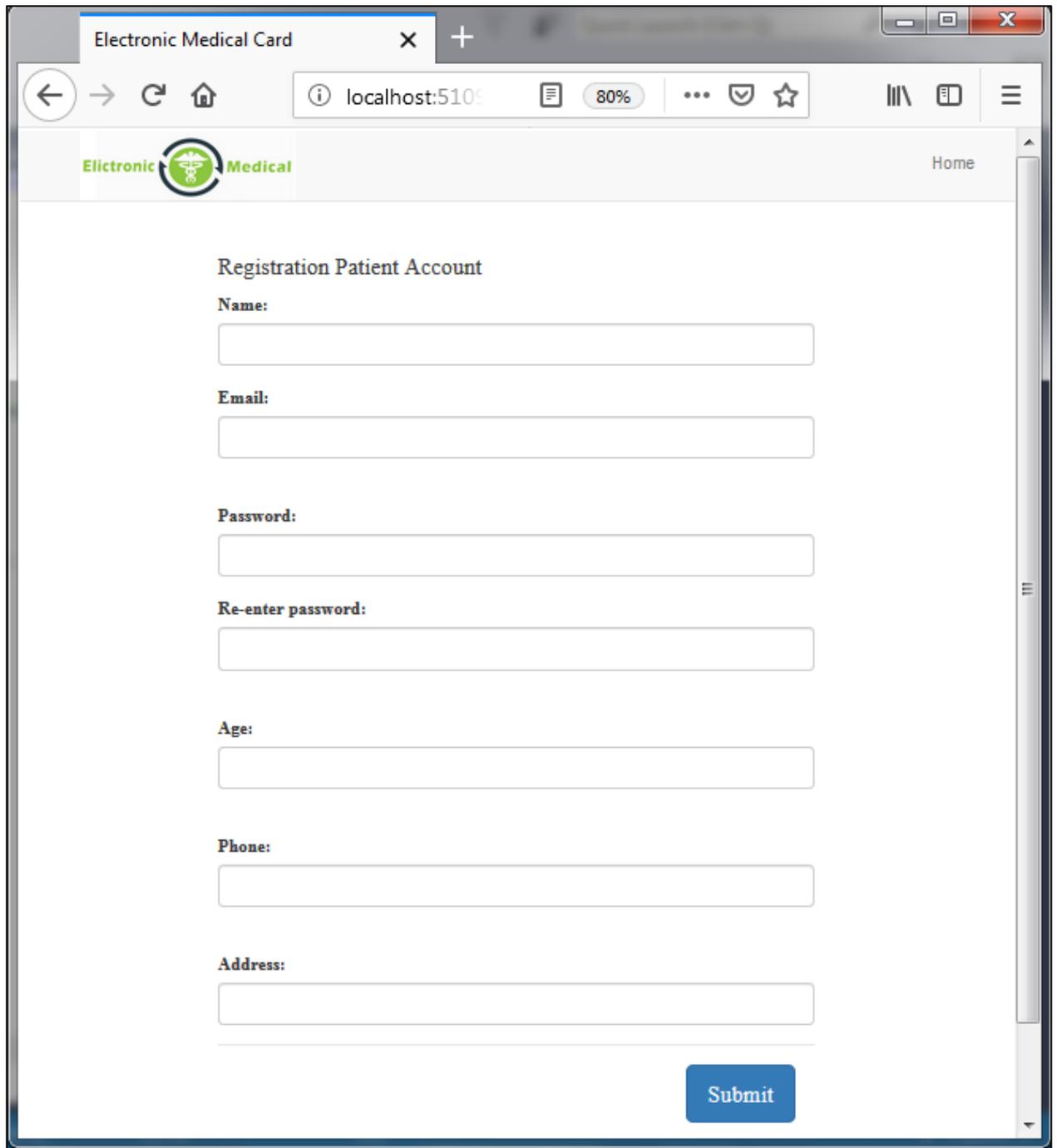
- Name:** A text input field.
- Email:** A text input field.
- Password:** A text input field.
- Re-enter password:** A text input field.
- Speciality:** A dropdown menu with the placeholder text "Select Speciality".
- Phone:** A text input field.

A blue "Submit" button is located at the bottom right of the form. The footer of the page contains the copyright notice "© 2019 Electronic Medical Card - Home" and a "Back to top" link.

Fig. 16. The page for the doctor register

The patient also can visit the main page can create new account by insert his information ("Name", "Email", "password", "Age", "phone", address).

He can do that by select "Create an account" in the sign in form as shown in fig. 17.



The image shows a web browser window with the title "Electronic Medical Card". The address bar displays "localhost:5109". The page content includes a header with the logo "Electronic Medical" and a "Home" link. The main section is titled "Registration Patient Account" and contains the following form fields:

- Name:
- Email:
- Password:
- Re-enter password:
- Age:
- Phone:
- Address:

A blue "Submit" button is located at the bottom right of the form.

Fig. 17. The page for the patient register

The doctor can make sign in to his special page by entering his username and password into the "Login" as shown in the fig. 18.

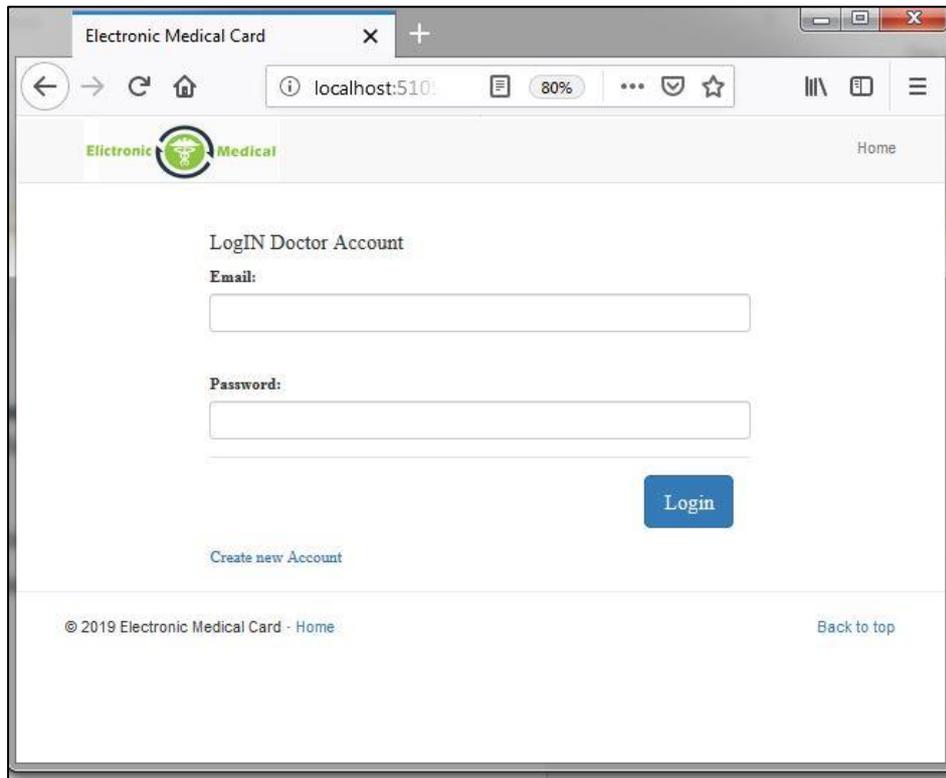


Fig. 18. The page for Doctor Login

The patient also can make sign in to his special page by entering his username and password into the "Login" as shown in the fig. 19.

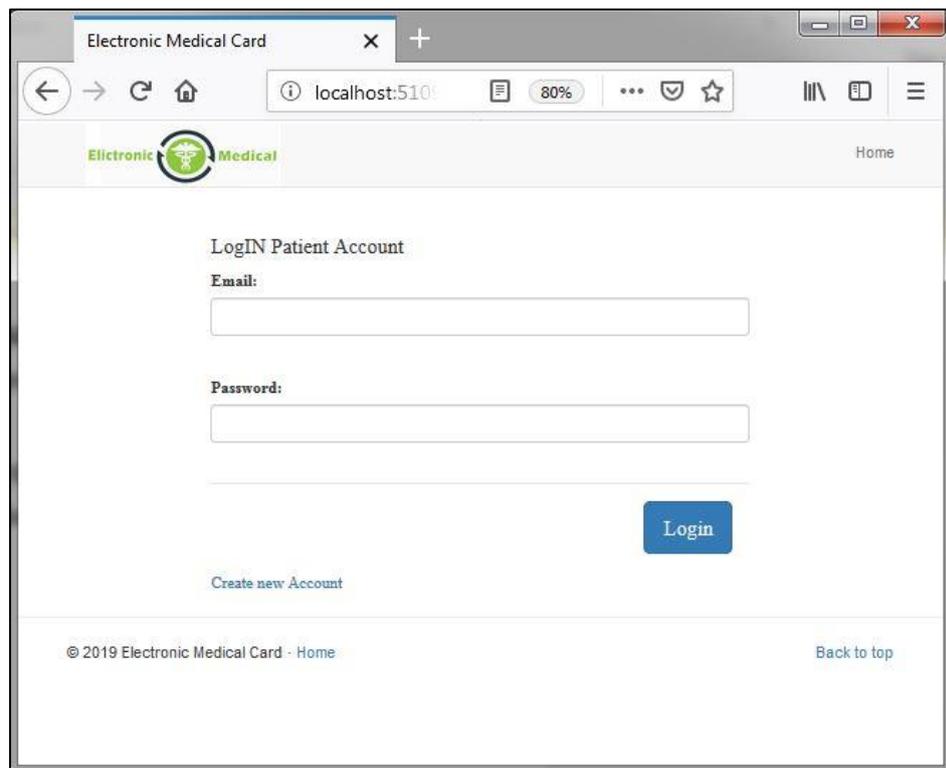


Fig. 19. The page for patient login

The doctor and after entering his page can edit his profile information as shown in fig. 20 or enter to the patient Review page as shown in fig. 21.

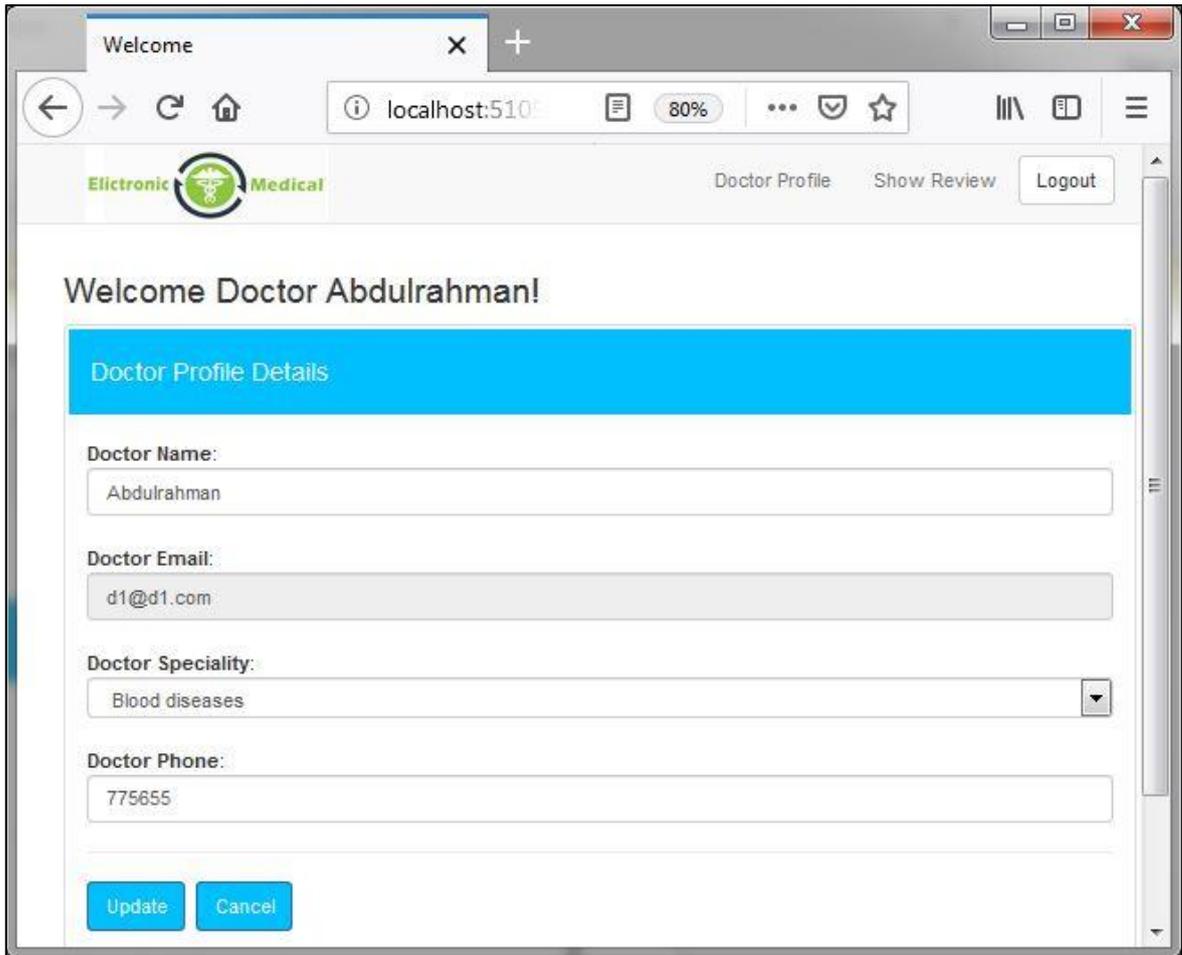


Fig. 20. Doctor Profile editing page

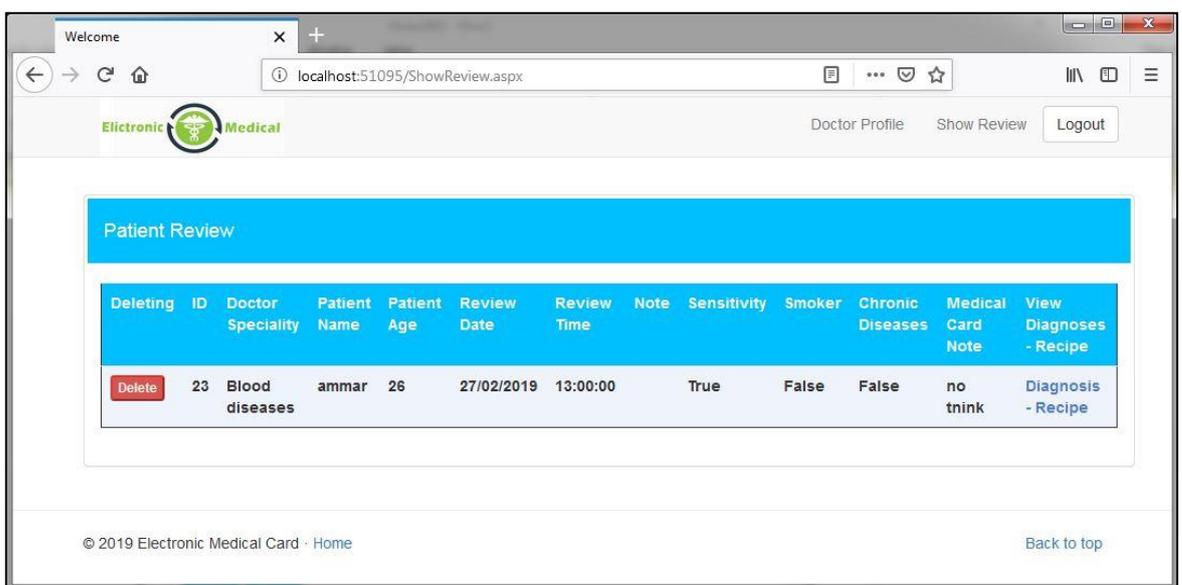


Fig. 21. Patient review page

The patient and after entering his page can (1) edit his profile information as shown in fig.22, (2) enter to medical card page as shown in fig.23, (3) or enter to the patient Review page as shown in fig.24.

The screenshot shows a web browser window titled "Electronic Medical Card" with the URL "localhost:51095/HomeP". The page features a navigation bar with "Patient Profile", "Medical Card", "Review", and "Logout" links. A blue header bar reads "Profile Details". Below it, the patient's name "ali" is entered in the "Patient Name" field, "assa12@gmail.com" in "Patient Email", "23" in "Patient Age", "77" in "Patient Phone", and "Baghdad" in "Patient Address". "Update" and "Cancel" buttons are at the bottom.

Fig. 22. Patient Profile editing page

The screenshot shows a web browser window titled "Electronic Medical Card" with the URL "localhost:51095/Medical". The page features a navigation bar with "Patient Profile", "Medical Card", "Review", and "Logout" links. A blue header bar reads "Medical Card". Below it, there are checkboxes for "Sensitivity", "Smoker", and "Chronic Diseases", all of which are checked. A "Note" field contains the text "No thing". An "Edit" button is located below the note field. The footer includes "© 2019 Electronic Medical Card - Home" and a "Back to top" link.

Fig. 23. Patient medical card page

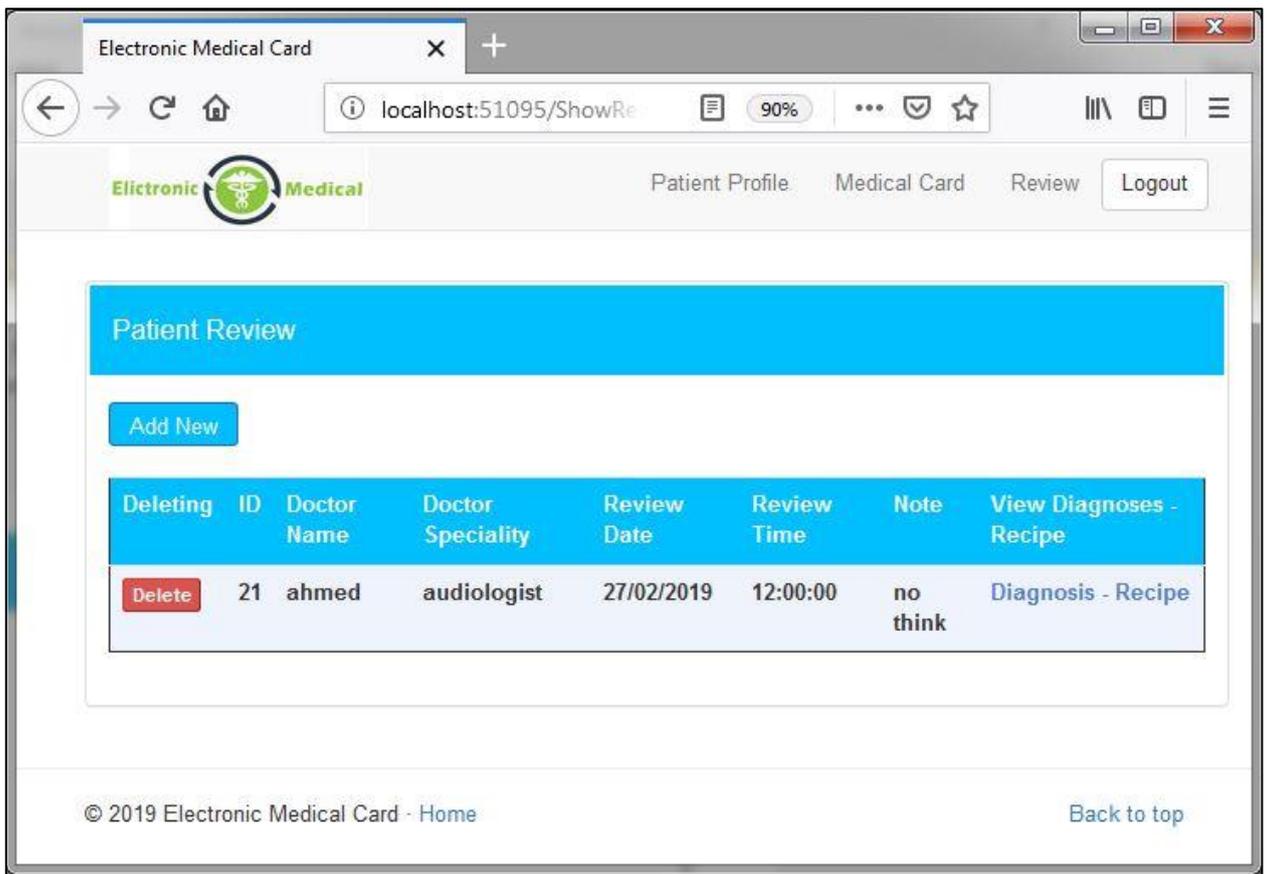


Fig. 24. Patient review page

4. TESTING OF WEB-APPLICATION

The Functional of the testing is used to check the functionality of the software application under test stage. Basically it is checking the basic functionality was mentioned in the functional specification document. Additionally, the testing of the software application is meeting to the user expectations. In another words it used to checking the behavior of the software application against test specification [20].

The kinds of software testing [21]: black box testing, white box testing, unit testing, incremental integration testing, integration testing, functional testing, system testing, end-to-end testing, sanity testing, regression testing, acceptance testing, load testing, stress testing, performance testing, usability testing, install-uninstall testing, recovery testing, security testing, compatibility testing, comparison testing, alpha testing, beta testing. Functional testing is a kind of testing which verifies that each function of the software application operates in conformance with the requirement specification.

This testing mainly involves black box testing and it is not concerned about the source code of the application. Each functionality of the system is checked by providing appropriate input data, verifying the output information and comparing the actual results with the expected results. This testing involved checking of User Interface, APIs, Database, security, client/server applications and functionality of the application under Test. The testing can be done by two options either the manually option or using automation running option [22].

Table 1 holds the results of testing for all functional requirements

Tab. 1. Results of functional testing

No	Function	Expected result	Obtained result	Conclusion
1.	To show the home page	Any guest can watch this page with the list of the main sections	Any guest can only watch this page with the list of the main sections	The function works
2.	To give all guests the	Any guest can see the page "Contact	Any guest can see the page "Contact	The function works

No	Function	Expected result	Obtained result	Conclusion
	permission to see the page "Contact us"	us" with the list of the main sections	us" with the list of the main sections	
3.	Register a new patient in the database of the registration page	The patient can insert the name, email, password, age, , phone, and address to register	The user can insert his first name, last name, mobile number, e-mail, date of birth, username, and password to register	The function works
4.	Register a new doctor in the database of the registration page	The doctor can insert his name, email, password, Specialty, and phone to register	The doctor can insert his name, email, password, Specialty, and phone to register	The function works
5.	The doctor has an on-site account	The doctor can make sign in to his special page by entering his username and password into the "Login"	The doctor can make sign in to his special page by entering his username and password into the "Login"	The function works
6.	The doctor has an on-site account	The patient can make sign in to his special page by entering his username and password into the "Login"	The patient can make sign in to his special page by entering his username and password into the "Login"	The function works

No	Function	Expected result	Obtained result	Conclusion
7.	Edit account	The doctor and after entering his page can edit his profile information or enter to the patient Review page a	The doctor and after entering his page can edit his profile information or enter to the patient Review page a	The function works
8.	Edit account	The patient and after entering his page can edit his profile information as shown in enter to medical card page or enter to the patient Review page	The patient and after entering his page can edit his profile information as shown in enter to medical card page or enter to the patient Review page	The function works

CONCLUSION

The initial goal of this study was to implement and develop an electronic medical card system. These services included health care, save and deliver care data to doctors. Our system has its own specific medical data saved in SQL server databases. All data has been managed by website application created by ASP.NET. The website application make the usage of this system easy and available for patients and doctors. The users can login and edit information from anywhere. It focuses on tasks directly related to patient care, unlike other healthcare information systems that support providers' and payers' operational processes.

The thesis therefore highlights the need to review most important report for patents and doctors. The reports was helped the doctors to make a good decision. The system classify as an information system with low power sources for environment. This environments designed by Microsoft like ASP.NET and SQL server.

The all statements are successfully implemented. First of all, we have been designed database by using SQL server as a management database system. We added many tables with necessary relational facilities such as primary key and foreign key. After that, we designed the website application by using ASP.NET in visual studio 2015 by using C# programming language. Then we integrated the database with website by using necessary classes and programming libraries. Finally, we have tested our system and run some operations such us view reports and edit information for system users.

For future work we suggest that this system can integrate with payment management system. Also add smart reader devices to read the information from card without need to type the card ID.

REFERENCES

1. Booch G., Rumbaugh J., Jacobson I. The Unified Modeling Language: Users Guide. – USA: Addison Wesley, 1998.490 – 512 p.
2. A Healthcare CFO's Guide to Smart Card Technology and Applications. // A Smart Card Alliance Healthcare Council Publication, February, 2009.
3. Additional information on smart card use in different vertical markets can be found on the Smart Card Alliance web site. [Electronic resource] URL: <http://www.smartcardalliance.org> (the date of access: 10.02.2019).
4. Chen, W.-M. & Chen, Y.-C., 2012. Web design and implementation for remote control. In Intelligent Control and Automation (WCICA). Beijing, 2012. IEEE.
5. Davis M.E., Phillips J.A. Learning PHP & MySQL: Step-by-Step Guide to Creating Database-Driven Web Sites. 2nd ed. – O'REILLY, 2007.
6. Novak R., Kandus G., Trcek D., Further development of a smart-card based health care information system in Slovenia. // Presented at the Fifth International Congress on Conference and Exhibition on Cards Applications in Health Care: Health Cards'99, Milan, Italy, 1999.
7. EMVCO, EMV2000 Integrated Circuit Card Specification for Payment Systems, BOOK 1 - Application Independent ICC to Terminal Interface Requirements, Version 4.0, December, 2000. [Electronic resource] IRL: <http://www.emvco.com/Specifications.cfm> (the date of access: 15.02.2019).
8. Mirzoev T, Sack L. WEBPAGE LOAD SPEED: ASP.NET VS. PHP. // I-manager's Journal on Information Technology, 2007. – No. 2(2). – P. 1–7.
9. Lancor, Katha S. Analyzing PHP frameworks for use in a project-based software engineering course. // 44th ACM technical symposium on Computer science education. New York, 2013. ACM.

10. Lucca G.A.D., Penta M.D., Antoniol G., Casazza G. An Approach for Reverse Engineering of Web-Based Applications. // Eighth Working Conference on Reverse Engineering, 2001. IEEE.
11. Meier J.D. et al. How To: Protect From SQL Injection in ASP.NET. – Microsoft Corporation, 2005.
12. Merlo E., Letarte D., Antoniol G. Automated Protection of PHP Applications Against SQL-injection Attacks. // Software Maintenance and Reengineering. Amsterdam, 2007. IEEE.
13. Moore A. User Input Validation in ASP.NET. – Microsoft Corporation, 2002.
14. Official site of Software Testing Class. [Electronic resource] URL: <http://www.softwaretestingclass.com/functional-testing/> (the date of access: 30.01.2019).
15. Official site of Software testing help. [Electronic resource] URL: <http://www.softwaretestinghelp.com/types-of-software-testing/> (the date of access: 29.01.2019).
16. Official site of Guru99. [Electronic resource] URL: <http://www.guru99.com/functional-testing.html> (the date of access: 01.02.2019).
17. Stacia V., Cherry D, D'Antoni J. Introducing Microsoft SQL Server 2016: Mission-Critical Applications, Deeper Insights, Hyper scale Cloud. Microsoft Press, 2016.180 - 199 p.
18. Jazayeri, M., 2007. Some Trends in Web Application Development. // Future of Software Engineering. Washington, 2007. IEEE Computer Society.
19. EMVCO, EMV2000 Integrated Circuit Card Specifications for Payment Systems, Book 4 - Cardholder, Attendant, and Acquirer Interface Requirements, Version 4.0, December, 2000. [Electronic resource] URL: <http://www.emvco.com/Specifications.cfm> (the date of access: 20.02.2019).
20. Garrett J. The Elements of User Experience: User-Centered Design for The Web and Beyond, Second Edition. – USA: New Riders, 2011.170. – 191 p.

21. Pesce J. Staunching Hospitals' Financial Hemorrhage with Information Technology. // Health Management Technology, August 2003. [Electronic resource] URL: <http://archive.healthmgttech.com/archives/h0803stanching.htm> (the date of access: 25.02.2019).

22. Cecchet E. et al. Performance comparison of middleware architectures for generating dynamic web content. // International Conference on Middleware. New York, 2003. Springer.