

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ
ФЕДЕРАЦИИ

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ

ВЫСШЕГО ПРОФЕССИОНАЛЬНОГО ОБРАЗОВАНИЯ

ТЮМЕНСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

ИНСТИТУТ МАТЕМАТИКИ И КОМПЬЮТЕРНЫХ НАУК

КАФЕДРА ИНФОРМАЦИОННЫХ СИСТЕМ

Отчет по дисциплине:

Программная инженерия

на тему: разработка программного приложения

«Поликлиника»

Выполнил:

студент 4 курса

группы ПИ 155-2

Шенгелия Д.Ю.

Проверил:

Красиков В.Е.

Содержание

Введение	3
Диаграмма IDEF1X (логическая часть).....	4
База данных	6
Разработка веб-службы	7
Реализация клиентского приложения.....	9
Навигация и действия с сущностями в форме.....	10
Заключение.....	14
Список литературы.....	14
Приложение 1. DoctorsController	15

Введение

Информационное общество подразумевает широкое применение компьютеров во всех сферах человеческой деятельности. В настоящее время в обществе огромную роль играют системы распространения, хранения и обработки информации, основанные на работе компьютера. Образуются межрегиональные и международные системы связи, которые позволяют обмениваться информацией на больших территориях за минимальные сроки. Наиболее известная такая система – интернет. Растет количество людей, профессионально занятых сбором, хранением и обработкой информации.

Еще одной тенденцией развития информационных технологий является глобализация информационного бизнеса. Чисто теоретически любой человек является сегодня потребителем информации. Поэтому возможности информационного рынка по-прежнему являются беспредельными, хотя и существует довольно жесткая конкуренция между основными производителями.

Современные информационные технологии — мощный инструмент, который в узком смысле обеспечивает конкурентоспособность предприятия и является возможностью совершенствования бизнеса. В широком смысле он показывает уровень развития страны, характеризуя степень ее вовлеченности в глобальные процессы. В связи с этим возникает необходимость оценки степени влияния информационных технологий (ИТ) на экономические процессы, то есть оценки их эффективности.

Объектом разработки является приложение «Поликлиника» в ASP.NET.

Диаграмма IDEF1X (логическая часть)

Методология моделирования IDEF1X предназначена для описания данных (информации). В ее основе лежит язык семантического моделирования, основанного на концепции "сущность — связь", позволяющей определять данные и связи между ними. Методология используется для создания информационной модели предметной области с помощью идентификации ее сущностей и связей между ними. Чаще всего такая методология используется для описания данных в целях последующей автоматизации их обработки с помощью систем управления базами данных [1]. Связи в IDEF1X представляют собой ссылки, соединения и ассоциации между сущностями. Связи — это глаголы, которые показывают, как соотносятся сущности между собой.

Диаграмма IDEF1x (см. рис. 1), разработанная для текущего проекта — утилизация отходов, показывает связи и отношения необходимые для разработки базы данных.

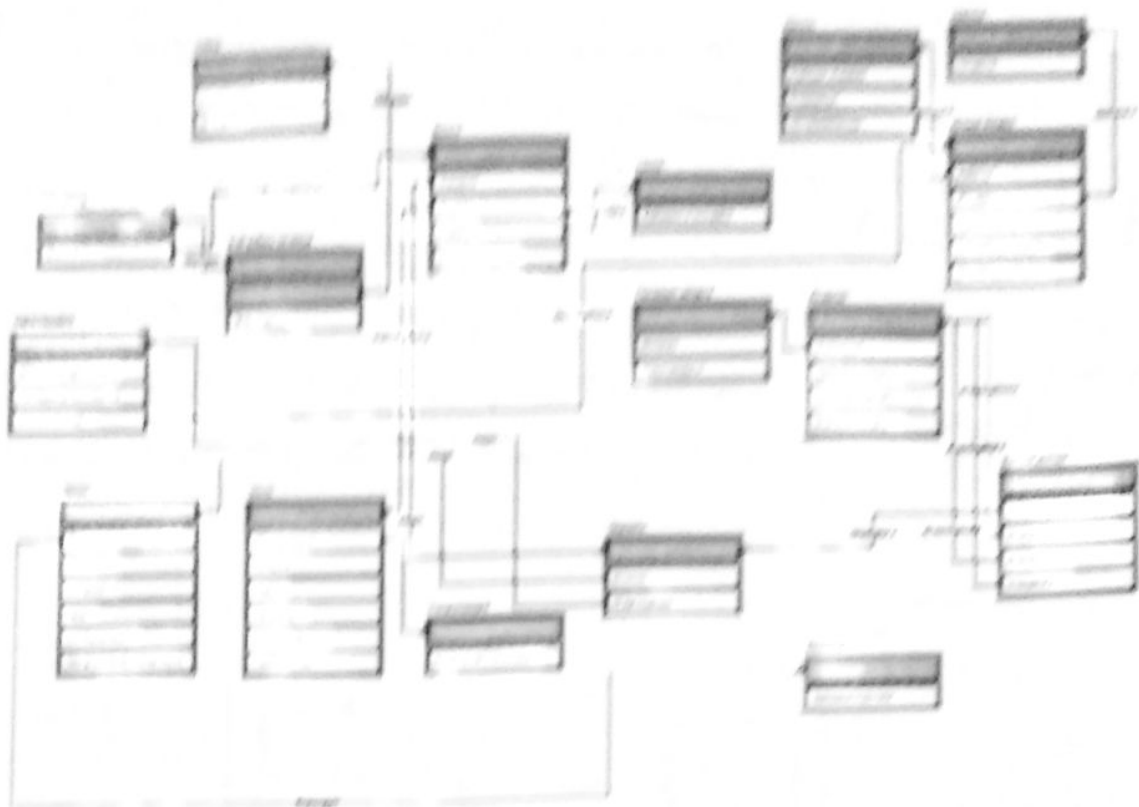


Рисунок 1. Диаграмма IDEF1X

После разработки диаграммы IDEF1X требуется создать базу данных с таблицами, соответствующими вышеописанным сущностям, с помощью специального программного обеспечения.

База данных

СУБД (Система управления базами данных) – это совокупность языковых и программных средств, предназначенных для создания, ведения и совместного использования БД многими пользователями. Система управления базами данных (СУБД) является посредником между базой данных и ее пользователями [2].

База данных (см. рис. 2), созданная в системе управления базами данных MS SQL, включает в себя 17 таблиц, описанных в диаграмме IDEF1X.

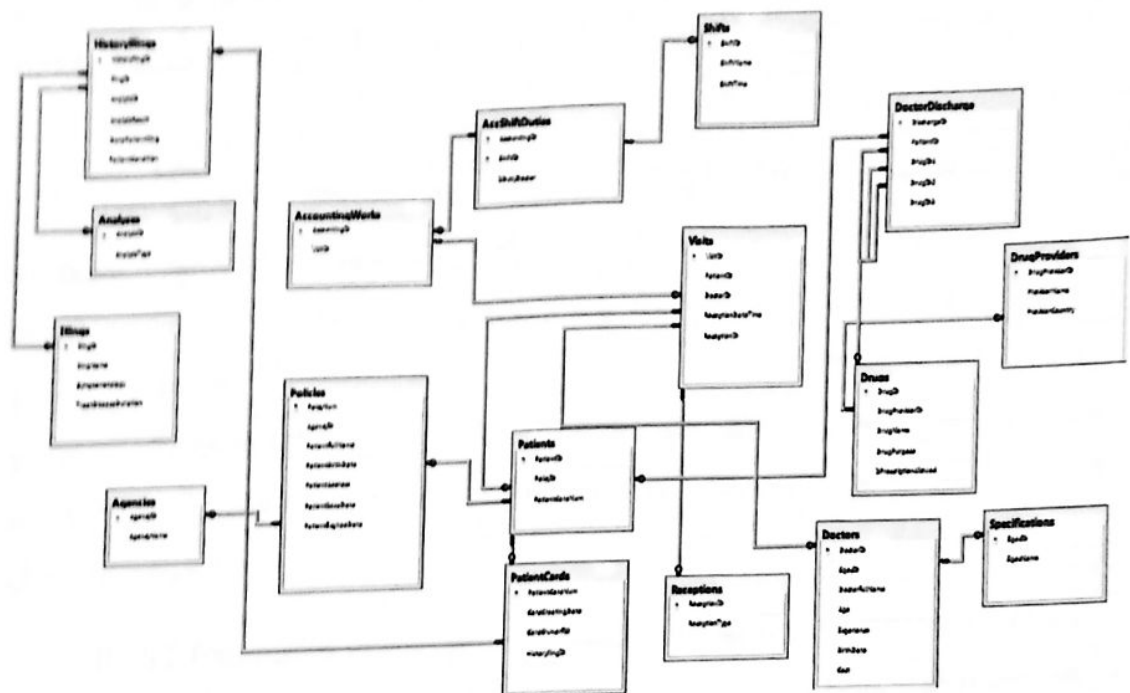


Рисунок 2. Диаграмма базы данных

После создания таблиц и добавления в них данных был выгружен скрипт с созданием таблиц и связей между ними с целью создания такой же структуры на другой базе.

Также были созданы хранимые процедуры (create, insert, update, delete), которые в дальнейшем будут использованы при создании представлений в проекте WebAPI MVC ASP.NET.

Разработка веб-службы

Для отделения бизнес-логики и логики работы с БД от клиентского приложения используются веб-службы, осуществляющие взаимодействие по REST.

Веб-служба, веб-сервис (англ. web service) — идентифицируемая уникальным веб-адресом (URL-адресом) программная система со стандартизированными интерфейсами, а также HTML-документ сайта, отображаемый браузером пользователя. Веб-службы могут взаимодействовать друг с другом и со сторонними приложениями посредством сообщений, основанных на определённых протоколах и соглашениях [3].

REST (сокращение от англ. Representational State Transfer — «передача состояния представления») — архитектурный стиль взаимодействия компонентов распределённого приложения в сети. REST определяет ряд архитектурных принципов проектирования Web-сервисов, ориентированных на системные ресурсы, включая способы обработки и передачи состояний ресурсов по HTTP разнообразными клиентскими приложениями, написанными на различных языках программирования. За последние несколько лет REST стала преобладающей моделью проектирования Web-сервисов [4].

Как происходит управление информацией сервиса — это целиком и полностью основывается на протоколе передачи данных. Наиболее

распространенный протокол – HTTP. Для HTTP действие над данными задается с помощью методов: GET (получить), PUT (добавить, заменить), POST (добавить, изменить, удалить), DELETE (удалить). Таким образом, действия CRUD (Create/Read/Update/Delete) могут выполняться как со всеми 4-мя методами, так и только с помощью GET и POST. Для каждой таблицы созданы контроллеры, в которых прописаны данные методы (приложение 1). Например, с помощью HTTP-запроса `http://localhost:58218/api/Doctors`, можно получить список всех врачей. При реализации в качестве возвращаемого значения было использовано визуальное представление сущности «Врачи» (см. рис. 3).

Врачи

новый записать

ФИО доктора	Возраст	Опыт	День рождения врача	Стоимость приема	Наименование специальности	
Жирков Юрий Дмитриевич	34.00	11.00	07.07.1985 0 00 00	40000.00	Детский терапевт	Редактировать Детали Удалить
Дмитриевская Надежда Михайловна	51.00	25.00	04.04.1967 0 00 00	15000.00	Психолог	Редактировать Детали Удалить
Салтыков Василий Петрович	42.00	19.00	11.03.1976 0 00 00	16650.00	Терапевт	Редактировать Детали Удалить
Дмитриевский Станислав Игоревич	33.00	7.00	06.06.1985 0 00 00	10000.00	Стоматолог	Редактировать Детали Удалить
Костылев Александр Александрович	61.00	20.00	01.03.1937 0 00 00	25000.00	Хирург	Редактировать Детали Удалить
Таликин Василий Дмитриевич	26.00	5.00	15.10.1990 0 00 00	6000.00	Детский терапевт	Редактировать Детали Удалить

Рисунок 3. Список всех врачей

Реализация клиентского приложения

В качестве инструмента для реализации клиентского приложения в веб-формате была использована технология ASP.NET MVC в совокупности с технологией Web API.

Слой приложения были распределены следующим образом:

- Model – в качестве модели сущности использовалась сущность, сгенерированная с помощью технологии Entity Framework.
- View – в качестве представления сущности был использован .cshtml-файл, сгенерированный на основе ранее созданных хранимых процедур.
- Controller – в качестве контроллера сущности был использован сгенерированный в режиме мастера создания контроллеров на основе Web API 2 (пример контроллера см. в приложении 1).

Таким образом были созданы контроллеры для всех сущностей, используемых в базе данных «Поликлиника».

В связи с тем, что в контроллерах в запросах (GET, POST и т.д.) в качестве возвращаемого значения, как правило, использовался тип ActionResult, который содержит в себе View-часть сущности, для окончательной реализации визуальной части оставалось лишь создать .aspx-форму для перенаправления на .cshtml-файлы сущностей.

В качестве формы главного меню была создана .aspx-форма (см. рис. 4) с кнопками, каждая из которых перенаправляет на .cshtml-файл соответствующей сущности.

Главное меню

Учеты работ

Учеты работ по смене

Агентства

Анализы

Выписки доктора

Врачи

Истории болезней

Болезни

Карты пациентов

Пациенты

Полисы

Приёмы

Смены

Специальности

Визиты

Рисунок 4. Главное меню БД «Поликлиника»

Навигация и действия с сущностями в форме

Рассмотрим этот пункт на примере нескольких сущностей («Врачи» и «История болезни»).

При переходе на любую из форм пользователю отображается список элементов сущности, которую выбрал пользователь, в нашем случае –

сущности

«Врачи»

(см.

рис.

5)

Врачи

сущность

ФИО доктора	Возраст	Опыт	День рождения врача	Стоимость приема	Специальность	
Жирков Юрий Дмитриевич	34.00	11.00	07.07.1992 0 00 00	40000.00	Детский терапевт	Редактировать Детали удалить
Дмитриевская Надежда Львовна	31.00	25.00	04.04.1987 0 00 00	15000.00	Психолог	Редактировать Детали удалить
Сидельников Василий Петрович	42.00	19.00	11.03.1976 0 00 00	18000.00	Терапевт	Редактировать Детали удалить
Дмитриевский Станислав Игоревич	33.00	7.00	06.06.1988 0 00 00	10000.00	Стоматолог	Редактировать Детали удалить
Колотилко Александр Алексеевич	51.00	26.00	01.03.1957 0 00 00	25000.00	Хирург	Редактировать Детали удалить
Тарасов Василий Дмитриевич	24.00	5.00	15.10.1999 0 00 00	60000.00	Детский терапевт	Редактировать Детали удалить

Рисунок 5. Список врачей

Пользователь может создать новую запись при помощи нажатия на одноименную гиперссылку, а также производить с записью следующий ряд действий:

- Редактировать – при нажатии на соответствующую гиперссылку пользователь переходит к форме редактирования записи (см. рис.

6)

Редактировать

Врач

Специальность	Детский терапевт
ФИО доктора	Жирков Юрий Дмитриевич
Возраст	26
Опыт	8
День рождения врача	07.07.1992 0 00 00
Стоимость приема	40000.00
Сохранить	

В список врачей

Рисунок 6. Редактирование данных по врачу

В результате редактирования была изменена запись (см. рис. 7).

Жирков Юрий Дмитриевич

26.00

6.00

07.07.1992 0 00 00

40000.00

Детский терапевт

Редактирование записи
Дата: 06.11.2018

После сохранения записи происходит переход к списку врачей. Если пользователь хочет отменить действие, он может перейти по гиперссылке «В список врачей».

- Детали – при нажатии на соответствующую гиперссылку пользователь переходит к форме, в которой показаны детали записи в строковом виде (в примере на рис. 8 показаны детали записи истории болезни)

Детали

История болезни

Результат анализа	Проблем не обнаружено
Дата болезни пацие...	03.11.2018 0 00 00
Состояние пациента	Нормальное
Тип анализа	Магнитно-резонансная томография
Название болезни	Ангина

Редактировать | В список историй болезней

Рисунок 8. Детали по записи истории болезни

В форме деталей пользователь может редактировать запись (более подробно см. в пункте «Редактировать»). Если пользователь хочет отменить действие, он может перейти по гиперссылке «В список истории болезней».

- Удаление – при нажатии на соответствующую гиперссылку пользователь переходит к форме удаления записи с предупреждением (в примере на рис. 9 показано удаление записи истории болезни)

Удалить

Уверены, что хотите удалить эту запись?

История болезни

Результат анализа	Гнойный отит
Дата болезни пацие...	04.12.2018 0:00:00
Состояние пациента	Удовлетворительное
Тип анализа	Рентген
Название болезни	Отит

| [В список историй болезней](#)

Рисунок 9. Форма удаления записи истории болезни

После удаления записи происходит переход к списку историй болезней. Если пользователь хочет отменить удаление записи, он может перейти по гиперссылке «В список историй болезней».

Заключение

В результате разработки программного приложения «Поликлиника» были выполнены следующие задачи:

- Проектирование базы данных «Поликлиника» в виде диаграммы IDEF1X.
- Создание базы данных «Поликлиника» в MS SQL Server, а также создание хранимых процедур (insert, update, create, delete) и создание скрипта в мастере сохранения скриптов MS SQL.
- Создание REST-сервиса с помощью технологии ASP.NET MVC + Web API.
- Создание форм сущностей в веб-формате.

Список литературы

1. https://studme.org/87187/ekonomika/metodologiya_ideflx
2. [https://ru.bmstu.wiki/СУБД_\(Система_Управления_Базами_Данных\)](https://ru.bmstu.wiki/СУБД_(Система_Управления_Базами_Данных))
3. <https://ru.wikipedia.org/wiki/Веб-служба>
4. <https://ru.wikipedia.org/wiki/REST>

Приложение 1. DoctorsController

```
using System;
using System.Collections.Generic;
using System.Data;
using System.Data.Entity;
using System.Linq;
using System.Net;
using System.Web;
using System.Web.Mvc;
using HospitalASP;

namespace HospitalASP.Controllers
{
    public class DoctorsController : Controller
    {
        private HospitalDBEntities db = new HospitalDBEntities();

        // GET: Doctors
        public ActionResult Index()
        {
            var doctors = db.Doctors.Include(d => d.Specification);
            return View(doctors.ToList());
        }

        // GET: Doctors/Details/5
        public ActionResult Details(Guid? id)
        {
            if (id == null)
            {
                return new HttpStatusCodeResult(HttpStatusCode.BadRequest);
            }
            Doctor doctor = db.Doctors.Find(id);
            if (doctor == null)
            {
                return HttpNotFound();
            }
            return View(doctor);
        }

        // GET: Doctors/Create
        public ActionResult Create()
        {
            ViewBag.SpecID = new SelectList(db.Specifications, "SpecID", "SpecName");
            return View();
        }

        // POST: Doctors/Create
        // Чтобы защититься от атак чрезмерной передачи данных, включите
        // определенные свойства, для которых следует установить привязку.
        // Дополнительные
        // сведения см. в статье http://go.microsoft.com/fwlink/?LinkId=317598.
        [HttpPost]
        [ValidateAntiForgeryToken]
        public ActionResult Create([Bind(Include =
            "DoctorID,SpecID,DoctorFullName,Age,Experience,BirthDate,Cost")] Doctor
            doctor)
        {
            if (ModelState.IsValid)
            {
                doctor.DoctorID = Guid.NewGuid();
                db.Doctors.Add(doctor);
                db.SaveChanges();
            }
        }
    }
}
```