

УДК 004

## ПРИМЕНЕНИЕ ITIL ПРИ РАЗРАБОТКЕ СИСТЕМ МОНИТОРИНГА ИСПРАВНОСТИ ОБОРУДОВАНИЯ

Кондратенков В. О., студент гр. ИТб-131, IV курс

Гаврилов А.А., студент гр.ИТб-131, IV курс

Научный руководитель: Ванеев О.Н., к.т.н. доцент

Кузбасский государственный технический университет имени Т.Ф. Горбачева  
г. Кемерово

Задача создания системы контроля неисправностей (Help Desk) на предприятии возникает уже не в первый раз, начиная с 80х годов были выработаны шаблоны создания такого рода приложений, один из подобных шаблонов описывается в библиотеке ITIL.

**ITIL** (*IT Infrastructure Library* – библиотека инфраструктуры информационных технологий) — библиотека, описывающая лучшие из применяемых на практике способов организации работы подразделений или компаний, занимающихся предоставлением услуг в области информационных технологий.

Библиотека ITIL включает в себя множество разделов основные из которых:

- **Поддержка услуг** (англ. *Service Support*) – представляет собой описание процессов, позволяющих обеспечить пользователям доступ к ИТ-услугам, необходимым для выполнения бизнес-задач.
- **Предоставление услуг** (англ. *Service Delivery*) – содержит описание типов ИТ-услуг, предоставляемых предприятием

Раздел **поддержки услуг** включает в себя:

- **управление инцидентами;**
- управление проблемами;
- управление конфигурациями;
- управление изменениями;
- управление релизами.

В нашем случае нас больше всего интересует управление инцидентами. **Инцидентом** считается любое событие, не являющееся частью нормального функционирования ИТ-сервиса. К инцидентам относятся, например, невозможность загрузить операционную систему, сбой электропитания, ошибка в программном обеспечении и т.д.

В соответствии с положениями библиотеки ITIL, основные функции которые должны быть реализованы при разработке систем мониторинга исправности оборудования:

- прием запросов пользователей;
- регистрация инцидентов и их категоризация;

- отслеживание развития инцидента;
- разрешение инцидентов;
- уведомление клиентов;
- закрытие инцидентов.

Получение запросов на обслуживание от пользователей может осуществляться различными способами: через интернет, через центр телефонной поддержки и т.п. Для упрощения обработки запросов клиентов рекомендуется создание единой точки обслуживания – специализированного приложения для поддержки пользователей Service Desk.

Заранее зная какие функции должна включать в себя программа, процесс ее создания значительно упрощается.

Все сводится к тому, что необходимо описать базу данных, и описать данные функции в том виде, который наиболее подходит для данного предприятия.

В качестве базового предприятия был выбран Кемеровский областной медицинский колледж (КОМК). На основе анализа объектов, необходимых для ведения процесса поддержки пользователей КОМК были выявлены следующие основные сущности:

- Персона (hPerson)
- Запрос (jQuery)
- Список работ(jWorkList)
- Журнал (jJournal)

Вспомогательные сущности:

- Уровень доступа (hAccess)
- Тип работы (hWorkType)
- Тип события (hEventType)
- Статус запроса (hState)
- Статус работы (hStateWork)
- Квалификация (hSkill)
- Квалификация\_Персона (jSkillList)

Реализация выделенных сущностей в виде базы данных представлена на рисунке 1.

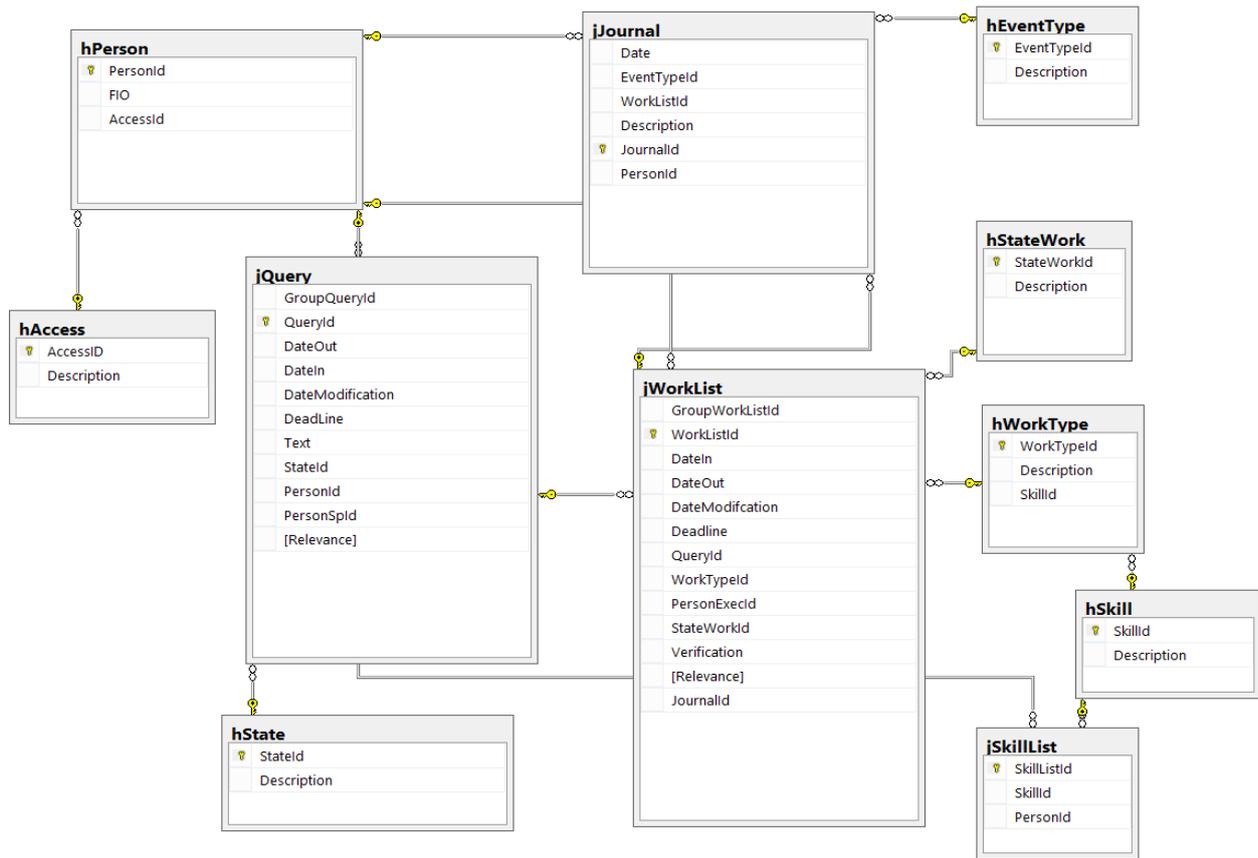


Рисунок 1. Схема БД

Далее требуется создать клиентскую часть. Для этого воспользуемся технологиями ASP.NET MVC и Entity Framework. Первая отвечает за визуализацию приложения в виде веб странички, вторая за связывание БД с приложением.

Прототип разработанного пользовательского интерфейса представлен на рисунке 2.

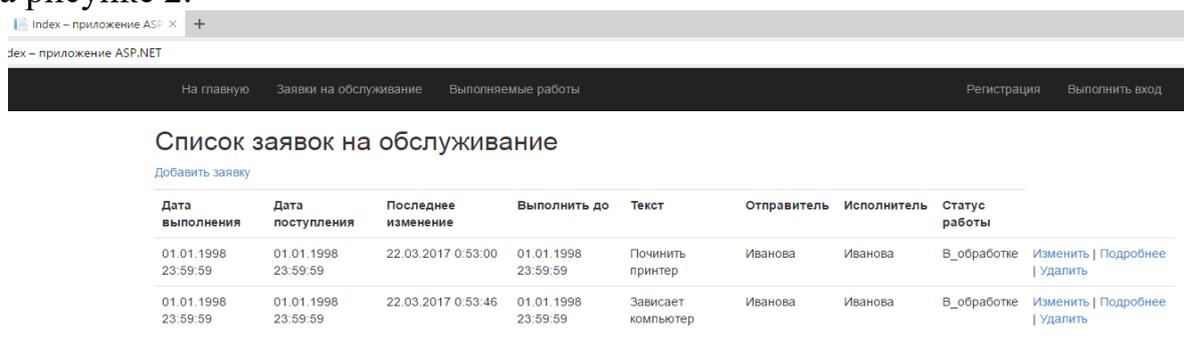


Рисунок 2. Прототип пользовательского интерфейса

Результатом связывания БД с приложением с использованием Entity Framework являются классы приложения, аналогичные таблицам БД, связи между таблицами (рисунок 3), будут также отображены на классах.

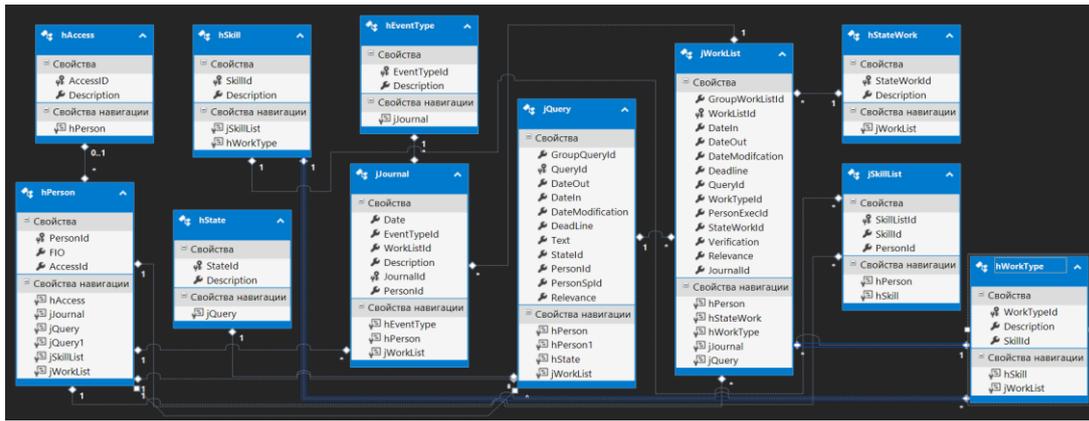


Рисунок 3. Созданные Entity Framework классы

Для дальнейшей работы требуется создать в коде приложения экземпляр контекста созданной модели БД, все изменения, сделанные в модели, будут отображены в БД. На рисунках 4 и 5 представлен код, отвечающий за

```

ModelKOMK.Context.cs  Index.cshtml  jJournalsController.cs  Edit.cshtml  Create...
Prak
using System.Data.Entity;
using System.Linq;
using System.Net;
using System.Web;
using System.Web.Mvc;
using Prak.Models;

namespace Prak.Controllers
{
    public class hPersonsController : Controller
    {
        private KOMK_Main_v2Entities db = new KOMK_Main_v2Entities();

        // GET: hPersons
        public ActionResult Index()
        {
            var hPerson = db.hPerson.Include(h => h.hAccess);
            return View(hPerson.ToList());
        }

        // GET: hPersons/Details/5
        public ActionResult Details(int? id)
        {
            if (id == null)
            {
                return new HttpStatusCodeResult(HttpStatusCode.BadRequest);
            }
            hPerson hPerson = db.hPerson.Find(id);
            //list<IQuery> iQuery =
    
```

отображение записей из БД, а также добавление новых записей.

```
// GET: hPersons/Create
ссылка: 0 | 0 исключений
public ActionResult Create()
{
    ViewBag.AccessId = new SelectList(db.hAccess, "AccessID", "Description");
    return View();
}

// POST: hPersons/Create
// To protect from overposting attacks, please enable the specific properties you want to bind to, for
// more details see http://go.microsoft.com/fwlink/?LinkId=317598.
[HttpPost]
[ValidateAntiForgeryToken]
ссылка: 0 | 0 исключений
public ActionResult Create([Bind(Include = "PersonId,FI0,AccessId")] hPerson hPerson)
{
    if (ModelState.IsValid)
    {
        db.hPerson.Add(hPerson);
        db.SaveChanges();
        return RedirectToAction("Index");
    }

    ViewBag.AccessId = new SelectList(db.hAccess, "AccessID", "Description", hPerson.AccessId);
    return View(hPerson);
}
```

При использовании знаний готовой библиотеки процесс разработки приложения значительно сокращается, снижается вероятность совершения ошибок на всех этапах разработки приложения.

Так мы вкратце рассмотрели один из основных моменты самой распространенной в мире на настоящий момент методики управления ИТ-сервисами: ITIL.

### Список литературы:

- 1) Электронный ресурс / <https://ru.wikipedia.org/wiki/ITIL>
- 2) Конспект лекций по дисциплине «Управление ИТ-сервисами и контентом» Составил: к.т.н., Бакаев Максим Александрович