

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ КЫРГЫЗСКОЙ  
РЕСПУБЛИКИ

КЫРГЫЗСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ  
УНИВЕРСИТЕТ

Им. И. РАЗЗАКОВА

# Отчет

по ПРЕДКВАЛИКАЦИОННОЙ ПРАКТИКЕ  
на тему Система обработки заявок «Help Desk»

Выполнил: ИСТТ(б)–1-16:

Максатбек уулу Элдияр

Руководитель работы:

Каримов Г.Т.

94/201)  
10.03.20

Бишкек 2020

# Содержание

## **1. Техническое задание**

- 1.1 Цели и задачи
- 1.2 Область применения
- 1.3 Основание для разработки
- 1.4 Назначение разработки
- 1.5 Требование к программе и программному продукту
- 1.6 Стадии и этапы разработки

## **2. Анализ предметной области**

- 2.1 Обзор систем обработки заявок

## **3. Реализация программы**

- 3.1. Обзор инструментов
- 3.2. Схема БД
- 3.3. Листинг программы
- 3.4. Интерфейс программы

ЗАКЛЮЧЕНИЕ

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

## Введение

Те, кто впервые слышат о системе автоматической обработки заявок, часто задают вопрос - service desk - что это и зачем нужно. Также многих руководителей интересует, как снизить уровень стресса у сотрудников техподдержки и повысить эффективность их работы.

Service Desk — своего рода диспетчерская служба, которая в полной мере ответственна перед клиентами или пользователями за предоставление согласованных с ними услуг, является центром приема всех жалоб и предложений, осуществляет контроль текущего состояния служб и имеет полномочия по выдаче нарядов на устранение возможных сбоев, а также на контроль процесса устранения неисправностей. В самом общем виде она может осуществлять перечисленные функции в отношении всех (а не только связанных с ИТ) услуг, предоставляемых организацией и ее отдельными подразделениями. Однако в дальнейшем будем рассматривать только аспект поддержки именно ИТ-услуг.

Из подобного определения не совсем очевидна необходимость такой службы. Остановимся на этом более подробно. В условиях постоянного усложнения используемых ИТ-услуг и возрастания требований к ним вопрос нормального функционирования ИТ-инфраструктур равноценен вопросу успешности бизнеса. Эффективное сопровождение используемых программ и оборудования оказывается критичным для достижения стоящих перед организацией бизнес-целей.

Когда пользователь или клиент сталкиваются с какой-либо проблемой (неисправностью, сбоем, просто неумением), он рассчитывает получить квалифицированную помощь в работе с приобретенной им услугой или продуктом. При этом его интересует максимально быстрое разрешение проблемы. Мало что раздражает больше, чем многократные звонки в некую организацию с целью найти там нужного человека — тем более, когда толком и не знаешь, а кто тебе, собственно, нужен.

К тому же, даже если такой сотрудник нашелся, он может быть занят каким-либо другим делом (например, решением проблемы другого клиента). Как быть в таком случае? Кто должен определить приоритеты и принять решение о порядке обслуживания? На каком основании? А как быть, если это происходит в нерабочее время? Можно задать много подобных вопросов. Как найти на них ответы?

Для устранения этих и многих других проблем и вводится Service Desk. Термин этот не является общепринятым; подобная структура может именоваться «Горячей линией» (Customer Hot Line), «Центром приема сообщений» (Call Center), «Центром технической поддержки» (Technical Support Center), «Диспетчерской помощью клиентам» (Help Desk) или каким-либо иным образом. Как правило, различие в наименовании скрывает в себе и некоторое различие в функциональности.

Каким образом Service Desk в состоянии помочь моему бизнесу? Что именно я получу, если потрачу время и деньги на создание такой службы? Не окажется ли она еще одним «нахлебником» в и без того довольно громоздкой ИТ-инфраструктуре?

Service Desk обеспечивает единую точку контакта для пользователей, клиентов, ИТ-персонала, ИТ-услуг и возможных «внешних» организаций, являющихся поставщиками каких-либо вспомогательных услуг (например, электропитания, внешних коммуникаций и т.д.). Для клиента это — наиболее важная (в стратегическом плане) функция ИТ-подразделения. Действительно, клиент, как правило, пользуется услугами без помощи сотрудников ИТ-подразделения, обращаясь к ним только в особых случаях (например, при

возникновении инцидента или при желании внести изменения в структуру получаемых услуг) и при этом общаясь именно с оператором Service Desk. В свою очередь, внутри ИТ-подразделения именно Service Desk отстаивает интересы клиента перед остальным персоналом.

Далее Service Desk несет ответственность за устранение инцидента. Даже если реально работы выполняются сотрудниками других подразделений (в том числе, и «внешними» партнерами), полный учет времени устранения инцидента, контроль за процессом устранения, информирование пользователя о состоянии проблемы, информирование руководства о задержках в устранении, их причинах — все это обязанности службы Service Desk. Как и предыдущая, эта функция в большей степени ориентирована на клиента и качество предоставляемых ему услуг. При этом не надо забывать, что службе Service Desk должны быть предоставлены соответствующие права.

Кроме этого, Service Desk формирует разнообразную управленческую информацию, в том числе об уровнях загруженности ресурсов; производительности и эффективности предоставляемых услуг; необходимости обучения клиентов; совокупной стоимости услуг; дефиците услуг и т.д.

Снижая стоимость владения ИТ-инфраструктурой путем повышения эффективности использования ресурсов, обеспечивая управление внесением изменений в ИТ-инфраструктуру и процессы в распределенной бизнес-среде, оптимизируя инвестиции и управляя функциями поддержки бизнеса, выявляя бизнес-возможности, помогая гарантировать удовлетворенность клиентов и сохранить их, Service Desk участвует в формировании прибавочной стоимости.

Не стоит думать, что Service Desk нужна только в крупных организациях со значительным количеством пользователей и услуг. Как показывает опыт, в небольших структурах управление ресурсами, технологиями и уровень документирования необходимых процедур и решений могут быть не менее важны.

## 1. Техническое задание

### 1.1 Цели и задачи

Цель - создать систему обработки заявок «help desk»

Задачи и функции , поставленные для реализации данной цели:

- Провести обзор предметной области
- Создать задачи для бизнес процесса
- Спроектировать бизнес процесс системы обработки заявок
- Разработать приложение

### 1.2 Область применения

Предполагаемыми пользователями данной системы могут быть, если ваша деятельность связана именно с предоставлением услуг:

2. ИТ – обслуживание
3. Вендорская или дилерская тех поддержка
4. Поддержка сайтов
5. Обслуживание промышленного оборудования, инженерных коммуникаций , систем видеонаблюдения и т.д.

### 1.3 Основание для разработки

Задача от начальника ОИТ ЗАО «КШБ»

Плановые сроки начала и окончания работы по созданию системы

- Начало 3.02.2020
- Окончание 06.03.2020

### 1.4 Назначение разработки

Внедрение Help Desk выгодно для ИТ подразделений и конечных пользователей. Подобные сервисы устраняют проблемы, которые связаны с медленной реакцией ИТ служб на запросы. Также повышается качество услуг организаций, которые имеют несколько офисов и подразделений в разных регионах. Соответственно, улучшается безопасность ИТ инфраструктуры компаний, уменьшаются затраты и появляется возможность их прогнозировать.

### 1.5 Требование к аппаратным средствам

Программное средство должно функционировать при минимальных требованиях к аппаратному обеспечению пользователя:

- Процессор Intel Pentium и выше

- ОЗУ 256 Mb
- VGA адаптер
- ОС Windows 7 и выше

Интерфейс программного средства должен быть дружелюбным и интуитивно понятным

## 1.6 стадии и этапы разработки

Состав работ включает четыре основных этапа

- планирование и анализ требований
- проектирование систем
- реализация системы обработки заявок
- оформление документации

### **Этап 1. Планирование и анализ требований**

Включает изучение документации, обзор существующих средств разработки системы обработки заявок, рассмотрение функций существующих систем, а также консультация со специалистами по различным вопросам данной предметной области

В данный этап так же должно входить работы по рассмотрению условий, при которых будет использоваться система обработки заявок для автоматизации ИТ услуг

### **Этап 2 . Проектирование системы**

На данном этапе разработки должно быть выполнены работы по моделированию функциональных требований к проектируемой системе и работы по разработке логической и физической модели данных

### **Этап 3. Реализация системы обработки заявок**

На данном этапе должно быть выполнены реализация интерфейса системы, отлажена функциональность

### **Этап 4. Оформление документации**

Данный этап включает работы по оформлению пояснительной записки к системе

## 2. Анализ предметной области

### 2.1 Обзор систем обработки заявок

#### Зарубежные системы

**OTRS** - Тикетная система с открытым кодом (*Open-source Ticket Request System*). Пользуется популярностью среди больших корпораций (Sony, HP, Яндекс и даже НАСА), во многом благодаря возможности вносить серьезные изменения в код и создавать на основе OTRS фактически собственную тикетную систему, адаптированную под специфические задачи.

Интерфейс заметно отличается от чисто коммерческих хелпдеск-сервисов – его можно назвать менее интуитивным, но удобным после должного изучения. Список тикетов отображается как в виде очереди заявок, так и в виде отдельных тикетов с их текущим статусом. Между двумя вариантами отображения необходимо переключаться. Доступно групповое редактирование тикетов, а также блокировка тикета сотрудником, взявшим его в работу. При этом можно указать время обработки заблокированного тикета, по истечении которого заявка вновь становится доступна для других сотрудников (если взявший не успел ее закрыть).

OTRS бесплатна, однако есть облачные бизнес-версии, в которые входит поддержка специалистов.

#### *Плюсы*

- Базовая версия полностью бесплатна.
- Наличие версий для Windows и Linux. Бесплатное дополнение позволяет управлять системой с iOS-клиента.
- Крайне широкие возможности по настройке и доработке системы.
- Система работает с протоколом LDAP и библиотекой ITIL.
- Подробная документация, основная часть которой переведена на русский язык.

#### *Минусы*

- Плата за гибкость и возможность переделывать систему под собственные задачи – сложность в интеграции, настройке и обслуживании. Необходимо иметь специалиста, который будет заниматься системой. Альтернативный вариант – приобретение платной версии с полноценной поддержкой от разработчиков.
- Интерфейс, к особенностям которого необходимо адаптироваться.

#### **Российские системы**

**VSDESK** - Простая и относительно молодая тикетная система от российских разработчиков. Интерфейс отличается лаконичностью – несмотря на множество опций, он не выглядит перегруженным. Тикеты создаются либо клиентами через веб-приложение, либо сотрудниками, принимающими звонки. При этом есть возможность создавать тикеты как в полной, так и в упрощенной форме, что экономит время клиента при возникновении несущественной проблемы или вопроса.

Существует три платные версии системы – «Старт», «Про» и «Корп», предназначенные для небольших, средних и крупных компаний. Для каждой версии установлен разовый платеж – 30, 60 и 90 т.р. Различаются версии длительностью техподдержки (которая при необходимости может быть продлена) и наличием дополнительных опций – SMS-уведомления, внутренний чат, отчеты и прочее в «Про», а также доступ к исходному коду и AD интеграция в «Корп».

### *Плюсы*

- Возможность скачать демо-версию с лимитом в 100 тикетов.
- Отсутствие ограничений по количеству пользователей и сотрудников.
- Лаконичный интуитивный интерфейс.
- Удобный процесс работы с тикетами.
- Возможность доработки при приобретении версии с исходным кодом.

### *Минусы*

- Нельзя изменять и гибко настраивать роли пользователей в версии «Старт».
- В процессе работы можно столкнуться с недочетами (которые достаточно быстро исправляются разработчиками).
- Самостоятельная доработка возможна только при приобретении версии «Корп» с исходниками в комплекте.

## **3. Реализация программы**

### 3.1 Обзор инструментов

**C#** (произносится си шарп) — объектно-ориентированный язык программирования. Разработан в 1998—2001 годах группой инженеров компании Microsoft под руководством Андерса Хейлсберга и Скотта Вильтаумота[7] как язык разработки приложений для платформы Microsoft .NET Framework. Впоследствии был стандартизирован как ECMA-334 и ISO/IEC 23270.

C# относится к семье языков с C-подобным синтаксисом, из них его синтаксис наиболее близок к C++ и Java. Язык имеет статическую типизацию, поддерживает полиморфизм, перегрузку операторов (в том числе операторов явного и неявного приведения типа), делегаты, атрибуты, события, свойства, обобщённые типы и методы, итераторы, анонимные функции с поддержкой замыканий, LINQ, исключения, комментарии в формате XML.

**ASP.NET MVC Framework** — фреймворк для создания веб-приложений, который реализует шаблон Model-view-controller.

В апреле 2009 года исходный код ASP.NET MVC был опубликован под лицензией Microsoft Public License (MS-PL)<sup>[2]</sup>. 27 марта 2012 года лицензия была изменена на Apache License 2.0<sup>[3]</sup>

В настоящее время разрабатывается ASP.NET MVC 6, как часть ASP.NET Core; 27 июня 2016 года состоялся выход версии 1.0.0<sup>[4]</sup>.

Платформа ASP.NET MVC базируется на взаимодействии трех компонентов: контроллера, модели и представления. Контроллер принимает запросы, обрабатывает пользовательский ввод, взаимодействует с моделью и представлением и возвращает пользователю результат обработки запроса.

Модель представляет слой, описывающий логику организации данных в приложении. Представление получает данные из контроллера и генерирует элементы пользовательского интерфейса для отображения информации.



## Движок представлений

---

Для управления разметкой и вставками кода в представлении используется движок представлений. До версии MVC 5 использовались два движка: Web Forms и Razor.

Начиная с MVC 5 единственным движком, встроенным по умолчанию, является Razor. Движок WebForms использует файлы .aspx, а Razor — файлы .cshtml и .vbhtml для хранения кода представлений. Основой синтаксиса Razor является знак @, после которого осуществляется переход к коду на языках C#/VB.NET<sup>[26]</sup>. Также возможно и использование сторонних движков. Файлы представлений не являются стандартными статическими страницами с кодом html, а в процессе генерации контроллером ответа с использованием представлений компилируются в классы, из которых затем генерируется страница html.

## Маршрутизация

---

При обработке запросов фреймворк ASP.NET MVC опирается на систему маршрутизации, которая сопоставляет все входящие запросы с определенными в системе маршрутами, которые указывают какой контроллер и метод должен обработать данный запрос. Встроенный маршрут по умолчанию предполагает трехзвенную структуру: контроллер/действие/параметр.

## ASP.NET MVC и ASP.NET Web Forms

---

ASP.NET MVC и ASP.NET Web Forms являются двумя родственными технологиями, в основании которых лежит одна платформа ASP.NET. И все же ASP.NET MVC имеет ряд преимуществ перед ASP.NET Web Forms<sup>[27]</sup>:

- разделение ответственности (отдельная разработка разных компонентов — контроллера, моделей, представлений)
- улучшенная тестируемость
- повышенная гибкость и настраиваемость под собственные нужды

**LINQ** (Language-Integrated Query) представляет простой и удобный язык запросов к источнику данных. В качестве источника данных может выступать объект, реализующий интерфейс IEnumerable (например, стандартные коллекции, массивы), набор данных DataSet, документ XML. Но вне зависимости от типа источника LINQ позволяет применить ко всем один и тот же подход для выборки данных.

Существует несколько разновидностей LINQ:

LINQ to Objects: применяется для работы с массивами и коллекциями

LINQ to Entities: используется при обращении к базам данных через технологию Entity Framework

LINQ to Sql: технология доступа к данным в MS SQL Server

LINQ to XML: применяется при работе с файлами XML

LINQ to DataSet: применяется при работе с объектом DataSet

Parallel LINQ (PLINQ): используется для выполнения параллельных запросов

**Microsoft SQL Server** — система управления реляционными базами данных (СУБД), разработанная корпорацией Microsoft. Основной используемый язык запросов — Transact-SQL, создан совместно Microsoft и Sybase. Transact-SQL является реализацией стандарта ANSI/ISO по структурированному языку запросов (SQL) с расширениями. Используется для работы с базами данных размером от персональных до крупных баз данных масштаба предприятия; конкурирует с другими СУБД в этом сегменте рынка.

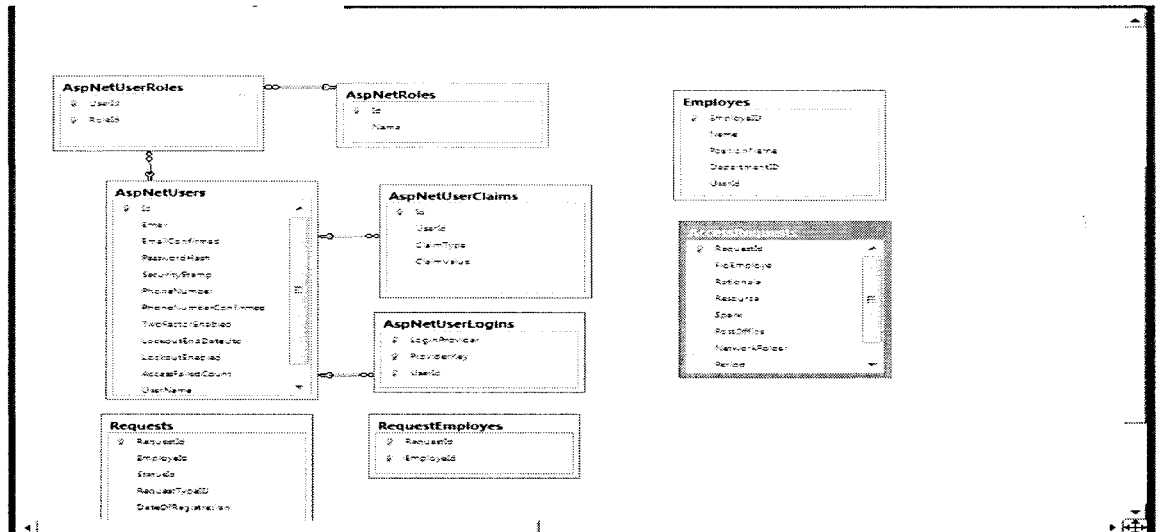
**Microsoft Visual Studio** — линейка продуктов компании Microsoft, включающих интегрированную среду разработки программного обеспечения и ряд других инструментальных средств. Данные продукты позволяют разрабатывать как консольные приложения, так и приложения с графическим интерфейсом, в том числе с поддержкой технологии Windows Forms, а также веб-сайты, веб-приложения, веб-службы как в родном, так и в управляемом кодах для всех платформ, поддерживаемых Windows, Windows Mobile, Windows CE, .NET Framework, Xbox, Windows Phone .NET Compact Framework и Silverlight.

Visual Studio включает в себя редактор исходного кода с поддержкой технологии IntelliSense и возможностью простейшего рефакторинга кода. Встроенный отладчик может работать как отладчик уровня исходного кода, так и отладчик машинного уровня. Остальные встраиваемые инструменты включают в себя редактор форм для упрощения создания графического интерфейса приложения, веб-редактор, дизайнер классов и дизайнер схемы базы данных. Visual Studio позволяет создавать и подключать сторонние дополнения (плагины) для расширения функциональности практически на каждом уровне, включая добавление поддержки систем контроля версий исходного кода (как, например, Subversion и Visual SourceSafe), добавление новых наборов инструментов (например, для редактирования и визуального проектирования кода на предметно-ориентированных языках программирования) или инструментов для прочих аспектов процесса разработки программного обеспечения

**SQL Server Management Studio (SSMS)** — утилита из Microsoft SQL Server 2005 и более поздних версий для конфигурирования, управления и администрирования всех компонентов Microsoft SQL Server. Утилита включает скриптовый редактор и графическую программу, которая работает с объектами и настройками сервера[1].

Главным инструментом SQL Server Management Studio является Object Explorer, который позволяет пользователю просматривать, извлекать объекты сервера, а также полностью ими управлять.

### 3.2 Схема БД



### 3.3 Листинг

#### Запросы на отображение данных через LINQ

```
public ActionResult Index()
{
    var allRequest = from request in db.Requests
                    join employe in db.Employees on request.EmployeeId equals employe.EmployeeID

    select new AllRequestViewModel
    {
        RequestId = request.RequestId,
        RequestType = request.RequestTypeID,
        StatusType = request.StatusId,
        EmployeeId = employe.Name,
        DateOfRegistration = request.DateOfRegistration,
        DateOfExit = request.DateOfExit
    };

    return View(allRequest.ToList());
}
```

```
Microsoft Visual Studio (Администратор)
Вид Проект Сборка Отладка Команда Средства Тест Анализ Окно Справка
Debug Any CPU BS Express (Google Chrome)
loyesController.cs AllRequestController.cs Report.cshtml HomeController.cs IBankController.cs EmployeeController.cs
Test HelpDeskTest.Controllers.AllRequestController Index()

    db.Requests.Add(request);
    // db.SaveChanges();
    return RedirectToAction("Index");
}

[Authorize(Roles = "Admin")]
public ActionResult Close(int? RequestId)
{
    if (RequestId == null)
        return HttpNotFound();
    Request request = db.Requests.Find(RequestId);
    if (request == null)
        return HttpNotFound();
    return View(request);
}

[HttpPost]
public ActionResult Close(Request request)
{
    var username = User.Identity.Name; // или username.Login
    var userID = db.Users.First(u => u.UserName == username).Id; // ID зарегистрированного пользователя
    var employee = db.Employees.First(e => e.UserId == userID); // сотрудник

    request.ExecutorId = employee.EmployeeID;
    var b = request.ExecutorId;
    ViewBag.Employee = b;
    request.DateOfRegistration = request.DateOfRegistration;
    request.DateOfExit = DateTime.Now;
    request.StatusId = StatusType.Finish;
    db.Entry(request).State = EntityState.Modified;
    db.SaveChanges();
    return RedirectToAction("Index");
}

public ActionResult Report(int? RequestId)
{
}

к Вывод
Строка 36 Столбец 10 Знак 10 ВСТ
```

```
Microsoft Visual Studio (Администратор)
Вид Проект Сборка Отладка Команда Средства Тест Анализ Окно Справка
Debug Any CPU BS Express (Google Chrome)
loyesController.cs AllRequestController.cs Report.cshtml HomeController.cs IBankController.cs EmployeeController.cs
Test HelpDeskTest.Controllers.AllRequestController Close(int? RequestId)

public ActionResult Report(int? RequestId)
{
    if (RequestId == null)
        return HttpNotFound();
    Request request = db.Requests.Find(RequestId);
    var employee = request.ExecutorId;
    var executor = request.EmployeeId;
    if (request == null)
        return HttpNotFound();

    Employee employeeName = db.Employees.FirstOrDefault(a => a.EmployeeID == employee);
    ViewBag.employee = employeeName;

    Employee executorName = db.Employees.FirstOrDefault(a => a.EmployeeID == executor);
    ViewBag.executor = executorName;

    var allExecutor = db.ConfirmRequestEmployees.Where(x => x.RequestId == RequestId).ToList();
    ViewBag.cx = executor;
    return View(request);
}

public ActionResult EditAllRequest(int? id)
{
    if (id == null)
        return HttpNotFound();

    Request request = db.Requests.Find(id);
    if (request.RequestTypeID == RequestType.IBankRequest)
        RedirectToAction("Details", "IBank", new { id });
    else
        return RedirectToAction("Create", "AccessRequest");
    return View();
}

к Вывод
Строка 68 Столбец 34 Знак 34 ВСТ
```

```
HelpDeskTest - Microsoft Visual Studio (Администратор)
Файл Проект Справка Отладка Команда Средства Тест Анализ Окно Сервис
Debug - Any CPU - IIS Express (Google Chrome) -
RequestEmployeesController.cs AllRequestControllers Report.cshtml EmployeeController.cs AccessRequestController.cs
HelpDeskTest
[ActionName]
public class AccessRequestController : Controller
{
    ApplicationDbContext db = new ApplicationDbContext();
    GET: action: Request
    public ActionResult Index()
    {
        var from = from request in db.Requests
        select accessrequest in db.AccessRequests on request.RequestId equals accessrequest.RequestId
        join employee in db.Employees on request.EmployeeId equals employee.EmployeeId
        select new SelectListItem()
        {
            RequestId = request.RequestId,
            EmployeeId = employee.Name,
            DateOfRegistration = request.DateOfRegistration,
            FIOEmployee = accessrequest.FIOEmployee,
            Rationale = accessrequest.Rationale,
            Spark = accessrequest.Spark,
            Resource = accessrequest.Resource,
            PostOffice = accessrequest.PostOffice,
            NetworkFolder = accessrequest.NetworkFolder,
            Period = accessrequest.Period,
        };

        return View(from.ToList());
    }

    GET: action: RequestDetails
    public ActionResult Details(int? id)
    {
        if (id == null)
        {
            return HttpNotFound();
        }
    }
}
90%
Список ошибок Вывод
Строка 148 Столбец 1 Знач 1 ВСТ
```

```
HelpDeskTest - Microsoft Visual Studio (Администратор)
Файл Проект Справка Отладка Команда Средства Тест Анализ Окно Сервис
Debug - Any CPU - IIS Express (Google Chrome) -
RequestEmployeesController.cs AllRequestControllers Report.cshtml AccessRequestController.cs UserRolesController.cs
HelpDeskTest
[ActionName]
public ActionResult EditUser(int? userID)
{
    ApplicationDbContext context = new ApplicationDbContext();

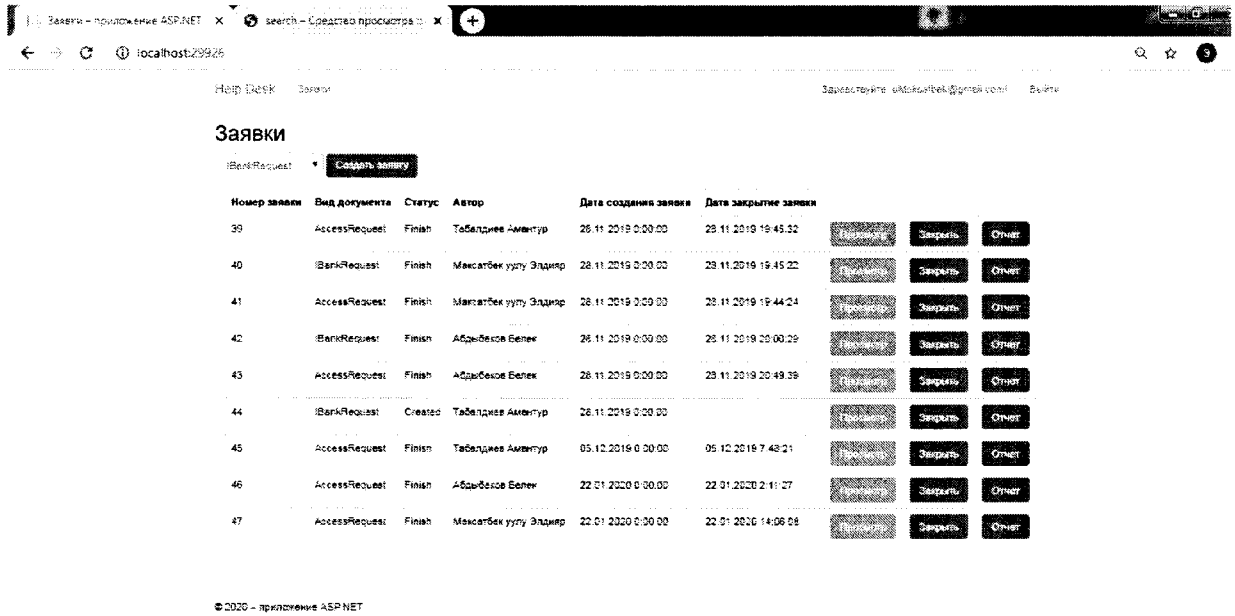
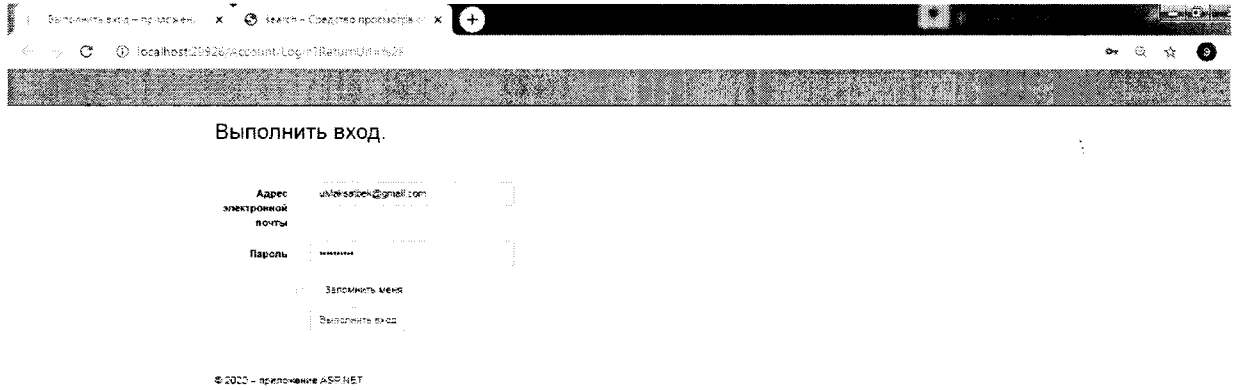
    var user = context.Users.FirstOrDefault(u =>
        u.Id == userID);

    var roles = context.Roles.ToList();
    var roleNames = roles.Select(r => r.Name).ToList();
    foreach (var userRole in user.Roles)
    {
        var role = roles.FirstOrDefault(r =>
            r.Id == userRole.RoleId);
        roleNames.Add(role.Name);
    }

    ViewBag.Roles = roles;
    var selector = new SelectListItem()
    {
        Value = user.Id,
        Text = user.Name
    }.ToList();

    return View(context.UserRoles.Where(u =>
        u.UserID == userID,
        u.UserName == user.UserName,
        u.RoleName == roles.Join(" ", roleNames)
    ));
}
90%
Список ошибок Вывод
Строка 12 Столбец 2 Знач 2 ВСТ
```

### 3.4 Интерфейс программы



localhost:25926/AccessRequest/Create

Help Desk Заявки

Администратор: oMakunbet@gmail.com Вышло

## Прошу предоставить доступ к следующим системам

Обоснование:

Срок:

Почта:

Сервисная сеть ВЦД:

Ресурс:

Период:

Принадлежность:

[Кликните здесь](#)

© 2020 – приложение ASP.NET

localhost:25926/AccessRequest/Report?RequestId=39

Help Desk Заявки

Администратор: oMakunbet@gmail.com Вышло

## Отчет

Заявка №. 39

Автор:

Таблица Амантур

	Статус	Finish
Тип документа	AccessRequest	
Дата создания	28.11.2018	
Дата закрытия	28.11.2018	

Исполнитель:

Максатбек уулу Оприлов

© 2020 – приложение ASP.NET

## Заключение

Многие службы поддержки начинали функционировать с обычных бумажных журналов учета, предполагающих индивидуализированную запись по каждому случаю, с отметкой о деталях и решениях. Однако таким образом можно только зафиксировать инциденты и сам факт их устранения. Большие возможности предоставляются компьютеризированными решениями, которые, помимо точности и аккуратности, одновременно дают возможность быстрого поиска и доступа к записям о ранее возникших ситуациях, известных ошибках, истории запросов от клиента и иной управленческой информации. Наибольшая польза достигается как раз от возможности легкого доступа к ранее практически недоступной информации.

Основные преимущества компьютеризированной реализации службы Service Desk:

доступность информации об инциденте всему персоналу службы поддержки;

сокращение периода обслуживания инцидента;

усовершенствованные процедуры отслеживания, эскалации и отработки инцидентов;

доступность в оперативном режиме более качественной информации (в том числе, об известных ошибках, решениях и истории запросов), а также внешних источников сведений;

большая доступность и точность управленческой информации;

устранение потерь, «забывчивости» и дублирования информации;

более качественное использование квалифицированного персонала;

облегчение решения совокупных задач и вычислений.

Современные системы Service Desk способны управлять, контролировать и отслеживать запросы на обслуживание, соблюдение условий контракта, людские ресурсы и последовательности работ. Эти системы интегрируются с остальными важными компонентами совокупной системы управления ИТ-ресурсами (в том числе с рекомендуемыми ITIL — Управлением изменениями, Конфигурированием и Учетом активов, Управлением ценой, Непрерывностью бизнеса, Планированием возможностей, Управлением сетями и т.д.).

Решающими факторами в выборе Service desk системы оказываются: возможность интеграции с остальными решениями по управлению различными элементами ИТ-инфраструктуры (сетями, серверами, рабочими станциями приложениями и т.д.); условия предоставляемой поддержки (ее полнота и доступность, а также возможность привлечения консультантов из компаний, специализирующихся на внедрении этих систем). При выборе необходимо обратить внимание на весь комплекс предстоящих расходов. Часто стоимость самого программного обеспечения существенно ниже стоимости обучения персонала и стоимости процедур внедрения. Необходимо учесть и перспективы дальнейшего развития системы.



## **СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ**

Герберт Шилдт: C# 4.0 полное руководство – 2011

<https://metanit.com/sharp/tutorial/15.1.php>

<https://metanit.com/sharp/mvc5/>

<https://www.w3schools.com/sql/default.asp>

[https://www.w3schools.com/bootstrap/bootstrap\\_ver.asp](https://www.w3schools.com/bootstrap/bootstrap_ver.asp)

КЫРГЫЗ РЕСПУБЛИКАСЫНЫН БИЛИМ БЕРҮҮ ЖАНА ИЛИМ МИНИСТРЛИГИ  
МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ КЫРГЫЗСКОЙ РЕСПУБЛИКИ

И.Разаков атындагы Кыргыз мамлекеттик техникалык университети

Кыргызский государственный технический университет им.И.Разакова

КҮНДӨЛҮК

ДНЕВНИК

практика боюнча

по практике Фидуквалитриккалмаша

студент \_\_\_\_\_

(Аты жөнү)

студента (ки) Максамбек уулу Т.И.Исра  
Ф.И.О.

тобу ИСТТЭ - 1-16 багыты (направление) 410200 ИСТ  
кесиби (специальность)

Институт Электромеханика и Техноколмушкаркалмаша  
факультети, институту (наименование факультета, института)

Практиканы өтүүчү жайы Кыргызско-Швейцарский Баши  
Мекеменин аталышы (наименование предприятия, организации)

Практиканын календарлык мөөнөтү  
Календарные сроки практики

Окуу планы боюнча башталышы " 10.02.20 " аягы " 6.03.20 "  
(По учебному плану начало) " \_\_\_\_\_ " конец " \_\_\_\_\_ "

Практикага келген мөөнөтү " \_\_\_\_\_ " \_\_\_\_\_ 20 ж.  
Дата прибытия на практику " 10 " февраль 20 20 г.

Практиканы аяктаган мөөнөтү " \_\_\_\_\_ " \_\_\_\_\_ 20 ж.  
Дата выбытия с места практики " 6 " март 2020 г.

Университеттен бекитилген жетекчи  
Руководитель от университета

Минбар \_\_\_\_\_ даража, кызматы

Кафедра ИСТТ звание, должность ит. преп

Аты жөнү \_\_\_\_\_

Фамилия Каримова Имя Гулмира

Отчество Токтомурамовна

Бишкек ш.  
г. Бишкек

Практикага тапшырма:

Задание на практику:

Адисттик боюнча

1. По специальности 710200 Информационные системы и технологии

2. Өндүрүштүк маркетинг жана экономика боюнча

По экономике и маркетингу прогизводства

ролетт стандартты ПТО, техника-экономикалык  
обоснование

3. Эмгекти коргоо боюнча

По охране труда

Ознакомление с ТБ и инструкциями на  
предприятии

4. Жеке тапшырма

Индивидуальное задание

- 1) Ознакомление со структурой организации
- 2) Посещение бизнес-процессов предприятия
- 3) Получение данных и изучение ЭДО
- 4) Постановка задачи
- 5) Проектирование информационной системы

Практиканы өтүү үчүн  
Күбөлүк  
Удостоверение  
на прохождение практики

Студенти \_\_\_\_\_

Студент(ка) Максамбек Уулу Э.  
(факультет, институт)

Багыты \_\_\_\_\_

Направление 710300 УСТ

Адистиги \_\_\_\_\_

Специальность Информационные сист. и технологии, БТМ

Топтор \_\_\_\_\_

Группы УСТ 5-1-1В

Иш сапары \_\_\_\_\_

Командируются в \_\_\_\_\_  
Ишкана, шаар Каргозеро-Швейцарский Банк, г. Бишкек

Практиканы өтүү үчүн \_\_\_\_\_

Для прохождения интервью практики \_\_\_\_\_

Мөөнөтү "10" феврале 2020 ж. "6" марты 2020 ж.

Буйрук № \_\_\_\_\_ от \_\_\_\_\_

Приказ № 5 нр / 101 от 14.01.2020



**ОИ боюнча проректор**  
Проректор по учебной работе

**Факультетин, институтун деканы**  
Декан факультета (института)

**Практиканы өтүү  
ГРАФИГИ**

Жуманын № № недели	Мөөнөтү Сроки	Аткарылган иштердин жана цехтин, участкастун кыскача мүнөздөө Цех, участок и краткая характеристика выполненных работ
1.	10.02.2020 - 14.02.2020	ЗАО „АШБ“ Знакомство с руководителем и составом коллектива
2	17.02.2020 - 21.02.2020	Разработка ТЗ Проектирование БД
3	24.02.2020 - 28.02.2020	Разработка программы
4.	2.03.2020 - 6.03.2020	Разработка интерфейса

**Практиканын жетекчилеринин колу**  
Подписи руководителей практики от:

Университеттен \_\_\_\_\_

колу

Университета \_\_\_\_\_

(ф.и.о. должность, подпись)

Ишканадан \_\_\_\_\_

колу

Предприятия \_\_\_\_\_

(ф.и.о. должность, подпись)



**Жумалык аткарылган иштердин жазылышы  
жана жетекчинин пикири**  
Еженедельная запись  
фактически выполненной работы и отзыв руководителя

Жума неделя	Мөөнөтү Сроки	Практиканын мазмуну Содержание прктики	Жетекчинин коруктундусу
1	10.02.2020	Эвюция	
	14.02.2020	интерактам	
		Ознакомление с деятельностью Башка	
2	17.02.2020	Разработка	
	21.02.2020	ТЗ для программы „Help Desk“	
		Проектиро- вание БЭ	
3	24.02.2020	Разработка	
	28.02.2020	программы	
4.	2.03.2020	Разработка	
	6.03.2020	интерфейса	

**ПРАКТИКАНЫН ЖАЛПЫ СУРООЛОРУ**  
**ОБЩИЕ ВОПРОСЫ ПРАКТИКИ**

Мадани й-массалык жана коомдук саясий, экскурсияларга катышуу  
(Участие в экскурсиях, общественно-политической и культурно-массовой работе)

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

Студенттин коомдук саясий жана маданий массалык иштерге катышуусуна  
ишкананын берген корутундусу  
(Заключение предприятия об участии студента (ки) в общественно-политических и  
культурно-массовых мероприятиях)

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

Ишкананын окулу

Представитель предприятия,  
Организации

(кызматы, колу) (должность, подпись)

Начальник ОУУ *[Signature]*



Практиканы өтүү туралуу  
Корутундук  
Заключение о прохождении практики

1. Ознакомился с деятельностью  
ЗАО "КИБ"
2. Участвовал в разработке ПО  
"Help Desk"
3. Вставлял ТЗ для ПО
4. Прокомментировал БД для ПО

Университеттин тарабынан практиканын жетекчиси  
(Руководитель практики от университета)

Өндүрүштөн  
(Производства)

Кафедрада практиканын өтүлүшүнүн жыйынтыгы каралган

(Отчет рассмотрен на кафедре)

" 10 " 03 2020 ж.

Баасы  
(Оценка)

Комиссия:

Душманов  
Кармалов  
Тойбаева