# Аннотация

*Авторы выпускной квалификационной работы*: Тартынский Алексей Валерьевич, Красильников Павел Витальевич

*Тема ВКР:* «Автоматизации процесса учёта деятельности на комбинате питания ГПБОУ «Кемеровский областной медицинский колледж», г. Кемерово». *Специальность:* Информационные системы и технологии.

*Город:* г. Кемерово.

*Год выполнения:* 2017.

*Количество страниц: 160*

*Количество таблиц:* 17

*Количество иллюстраций:* 84

*Описание:* Данная работа описывает разработку информационной системы по автоматизации процесса учёта деятельности на комбинате питания. Система функционирует на базе СУБД SQL Server. В работе рассматриваются вопросы: аппаратной и административной интеграция, общие вопросы администрирования, вопросы информационной безопасности, разработка системы централизованного хранения и обработки данных, технология разработки и программной реализации, расчет экономической эффективности. Особенностью системы является автоматическая генерация меню на каждый день.

Содержание

[Аннотация 2](#_Toc485006303)

[Введение 6](#_Toc485006304)

[1 Разработка технического задания 7](#_Toc485006305)

[1.1 Исходное задание на проектирование 7](#_Toc485006306)

[1.2 Анализ организационно-технологической структуры предприятия 7](#_Toc485006307)

[1.2.1 Название организации, статус организации, форма юридического лица 7](#_Toc485006308)

[1.2.2 Масштаб организации 8](#_Toc485006309)

[1.2.3 Общие принципы функционирования организации 8](#_Toc485006310)

[1.2.4 Вид и профиль деятельности 8](#_Toc485006311)

[1.2.5 Цели функционирования организации 9](#_Toc485006312)

[1.2.6 Организационная структура предприятия 10](#_Toc485006313)

[Организационная структура Кемеровского областного медицинского коллежа приведена в приложении 2. 10](#_Toc485006314)

[1.2.7 Обследование функций, структуры и деятельности базового подразделения. 17](#_Toc485006315)

[1.2.8 Выявление бизнес-процессов базового подразделения. 21](#_Toc485006316)

[1.2.9 Анализ существующей реализации базового бизнес-процесса. 26](#_Toc485006317)

[1.3 Формулировка требований к системе 27](#_Toc485006318)

[1.3.1 Определение деятельностей, требующих автоматизации, кандидатов в требования. 27](#_Toc485006319)

[1.3.2 Определение состава функциональных требований к системе. 31](#_Toc485006320)

[1.3.3 Разработка сценариев, реализующих требования к системе. 36](#_Toc485006321)

[1.3.4 Определение требований к интерфейсу. 43](#_Toc485006322)

[1.3.5 Окончательная формулировка требований к системе. 48](#_Toc485006323)

[2 Анализ и проектирование 55](#_Toc485006324)

[2.1 Определение классов анализа 55](#_Toc485006325)

[3 Разработка системы централизованного хранения и обработки данных 71](#_Toc485006326)

[3.1 Формирование требований к БД 71](#_Toc485006327)

[3.2 Формирование отношений БД 72](#_Toc485006328)

[3.3 Даталогическое проектирование 73](#_Toc485006329)

[3.4 Разработка сценариев работы с данными 79](#_Toc485006330)

[3.5 Реализация сценариев работы с данными 81](#_Toc485006331)

[4 Специальная часть. 85](#_Toc485006332)

[4.1 Модульное тестирование приложений. 85](#_Toc485006333)

[4.2 Автоматизация процесса тестирования с использованием веб-сервиса AppVeyor. 89](#_Toc485006334)

[5 Технологии разработки и программная реализация 94](#_Toc485006335)

[5.1 Выбор операционной системы 94](#_Toc485006336)

[5.2 Выбор типа взаимодействия пользователя с системой, технологии пользовательского интерфейса 94](#_Toc485006337)

[5.3 Выбор технологии взаимодействия пользовательских компонент с данными 94](#_Toc485006338)

[5.4 Выбор языка и среды программирования 95](#_Toc485006339)

[5.5 Определение состава компонент 95](#_Toc485006340)

[5.6 Разработка схемы развертывания 96](#_Toc485006341)

[5.7 Требования к системе 96](#_Toc485006342)

[5.8 Разработка компонент 97](#_Toc485006343)

[5.9 Характеристика разработанного решения 107](#_Toc485006344)

[5.10 Разработка вопросов тестирования 108](#_Toc485006345)

[6 Аппаратная и административная интеграция ИС 113](#_Toc485006346)

[6.1 Разработка схемы развертывания 113](#_Toc485006347)

[6.2 Формулировка требований к физическим устройствам и сетевому оборудованию, состав рабочих мест 114](#_Toc485006348)

[6.3 Выбор состава аппаратных средств 114](#_Toc485006349)

[6.4 Расчет потребности персонала. 115](#_Toc485006350)

[6.5 Выбор сетевой архитектуры и технологии 115](#_Toc485006351)

[6.5.1 Выбор архитектуры 115](#_Toc485006352)

[6.5.2 Выбор технологии и аппаратных средств. Расчет сети 116](#_Toc485006353)

[7 Общие вопросы администрирования 118](#_Toc485006354)

[7.1 Определение стратегии администрирования на уровне руководства. 118](#_Toc485006355)

[7.2 Определение объектов администрирования на уровне предприятия. 118](#_Toc485006356)

[7.3 Политика администрирования на уровне предприятия. 118](#_Toc485006357)

[7.4 Определение объектов администрирования на уровне разрабатываемой ИС. 119](#_Toc485006358)

[7.5 Политика администрирования на уровне разрабатываемой ИС. 119](#_Toc485006359)

[8 Вопросы информационной безопасности 122](#_Toc485006360)

[8.1 Анализ угроз 122](#_Toc485006361)

[8.2 Информационная безопасность на уровне предприятия. 124](#_Toc485006362)

[8.2.1 Контроль доступа в помещения. 124](#_Toc485006363)

[8.2.2 Обеспечение безопасности с помощью аппаратных средств. 124](#_Toc485006364)

[8.2.3 Обеспечение безопасности с помощью программных средств. 124](#_Toc485006365)

[8.2.4 Определение политики управления доступом пользователей. 125](#_Toc485006366)

[8.2.5 Требования к административным обязанностям. 125](#_Toc485006367)

[8.2.6 Антивирусная защита информации. 126](#_Toc485006368)

[8.2.7 Информационная безопасность на уровне АИС. 127](#_Toc485006369)

[8.2.8 Защита персональных данных. 127](#_Toc485006370)

[9 Экономическое обоснование разработки ИС 129](#_Toc485006371)

[9.1 Расчет затрат на разработку системы 129](#_Toc485006372)

[9.2 Расчёт эксплуатационных затрат 134](#_Toc485006373)

[9.3 Экономическая эффективность 138](#_Toc485006374)

[9.4 Срок окупаемости 139](#_Toc485006375)

[9.5 Технико-экономические показатели проекта 139](#_Toc485006376)

[Список используемой литературы 142](#_Toc485006377)

[Заключение 143](#_Toc485006378)

[ПРИЛОЖЕНИЕ 1. Схема ЛВС 144](#_Toc485006379)

[ПРИЛОЖЕНИЕ 2. Организационная структура колледжа 145](#_Toc485006381)

[ПРИЛОЖЕНИЕ 3. Схема управления организационными структурами КОМК 146](#_Toc485006382)

[146](#_Toc485006383)

# 

# Введение

Системы, используемые на предприятии, за длительный период своего развития прошли путь от простейших форм учета до современных автоматизированных систем с использованием компьютеров.

Каждый работник предприятия, будь то, экономист, оператор, или бухгалтер, имеет собственное рабочее место с персональным компьютером. Соответственно его работа напрямую зависит от технического состояния его компьютера. Важность проведения своевременных профилактических работ с ПК, его ремонта или замены комплектующих на более современные и отвечающие требованием развивающихся информационных технологий, неоспорима. Отсутствие своевременного обслуживания может закончиться остановкой работы учреждения, следовательно, поиск информации по требуемому оборудованию необходимо осуществлять в ограниченные сроки. Материально ответственным лицам необходимо постоянно контролировать состояние оборудования, его месторасположение, перемещение, учитывать срок полезного использования и другие характеристики.

Таким образом, проблема создания и внедрения системы учёта средств вычислительной техники на предприятие ПАО «Ростелеком» является актуальной.

В ходе разработки системы были использованы технологии MS SQL Server 2012, MS Visual Studio 2012, позволившие обеспечить возможность оперативного и удобного получения от системы справочной информации, предоставили возможность для накопления и редактирования поступающих данных.

# Разработка технического задания

## Исходное задание на проектирование

Требуется автоматизировать процесс формирования меню и поставок.

В системе присутствуют следующие пользователи: сотрудник кухни, сотрудник склада, менеджер, администратор.

Формирование ежедневного меню с ценами. Цена блюда складывается из стоимости ингредиентов и человеко-часов на приготовление блюда. Все составляющие цены блюда могут меняться каждый день. Необходима возможность расчета и просмотра цены блюда.

## Анализ организационно-технологической структуры предприятия

### Название организации, статус организации, форма юридического лица

Целевым предприятием является государственное бюджетное профессиональное учреждение «Кемеровский областной медицинский колледж» (далее КОМК). Официальная дата открытия колледжа – 1995 год, данная организация является юридическим лицом.

Информация об учредителе:

В соответствии с п.1.7 Устава учредителем ГБПОУ «Кемеровский областной медицинский колледж» является Кемеровская область.

Функции и полномочия учредителя Колледжа от имени Кемеровской области осуществляет департамент охраны здоровья населения Кемеровской области.

Полное наименование учредителя: Департамент охраны здоровья населения Кемеровской области (сокращенно ДОЗН КО).

Адрес: 650064, Кемеровская область, г. Кемерово, пр. Советский, 58

Руководство: начальник департамента охраны здоровья населения Кемеровской области Шан-Син Владимир Михайлович.

### Масштаб организации

КОМК является крупной организацией. Штат головного учреждения насчитывает 650 человек. Годовой доход КОМК составляет 243,872 млн. рублей. Колледж имеет филиалы в других городах. В их числе:

* Анжеро-Судженский филиал (654041, Кемеровская область, г. Анжеро-Судженск, ул. Ленина, 18а);
* Беловский филиал (652600, Кемеровская область, г. Белово, ул. Юности,19);
* Мариинский филиал (652154, Кемеровская область,г. Мариинск,ул. Добролюбова, д.1.);
* Ленинск-Кузнецкий филиал (652515, Кемеровская область, г. Ленинск-Кузнецкий, ул. Зварыгина, д. 6);
* Междуреченский филиал (652877, Кемеровская область, г. Междуреченск, ул. Кузнецкая, д. 39);
* Новокузнецкий филиал (654041, Кемеровская область, г. Новокузнецк, ул. Кузнецова, 33).

### Общие принципы функционирования организации

Деятельность КОМК проводится в соответствии со следующими общими принципами функционирования организаций:

* - Скалярный принцип (scalaris - ступенчатый) - иерархичность построения организаций.
* - Принцип функциональности - должностные обязанности на каждой ступени управления определяются настолько детально, насколько возможно.
* - Принцип диапазона контроля – руководитель умственной деятельности имеет не более 8 человек.
* - Принцип личной ответственности - ответственность руководителя за действия подчинённых ему людей в полном объёме.

### Вид и профиль деятельности

Основная деятельность организации включает:

1. Обучение абитуриентов по специальностям:

* + «Лечебное дело», профиль **фельдшер**;
  + «Акушерское дело», профиль **акушерка**;
  + «Сестринское дело», профиль **мед. сестра/брат;**
  + «Лабораторная диагностика», профиль **медицинский лабораторный техник**;
  + «Фармация», профиль **фармацевт**;
  + «Стоматология», профиль **зубной техник**.

2. Предоставление услуг по повышению квалификации младшего мед. персонала и людей смежных профессий (МЧС, Пожарные службы).

Цели функционирования организации

Основной целью КОМК является выпуск дипломированных специалистов по указанным выше специальностям на выделенные бюджетные средства. Ежегодный выпуск организацией составляет около 800 молодых специалистов. В дальнейшем планируется увеличение числа выпускников до 900 в год. В следствие этого рассчитывается убедить государственный аппарат в выделении большего объема бюджетных средств на развитие организации.

Не менее важной целью является повышение квалификации уже состоявшихся работников здравоохранительных структур, их обучение современным практикам и технологиям.

Для поддержки выполнения основных целей функционирования, выделяется отдельная цель: Обеспечение бесперебойной работы колледжа - поддержание колледжа в работоспособном режиме.

Для разработки модели «Цели функционирования предприятия» используется диаграмма классов (class diagram), которая представлена на рисунке 1.1.

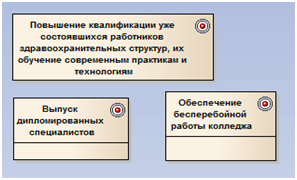


Рисунок 1.1 Цели функционирования КОМК

Организационная структура предприятия

## Организационная структура Кемеровского областного медицинского коллежа приведена в приложении 2.

#### Форма отчета о подразделениях предприятия

Таблица 1.1

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Код подразделения | Наименование подразделения | Выполняемые функции | Функциональная подчиненность | Структурная подчиненность |
| Д | Дирекция | Общее руководство КОМК | - | - |
| АХР | Отдел АХР | Материально-техническое обеспечение | Д | Д |
| УИМР | Управление учебно-методической работы | Руководство обучающим процессом | Д | Д |
| УДО | Управление доп. образования | Руководство процессами переподготовки специалистов и повышения квалификации | Д | Д |

Разработка ПО, а также обслуживание технических средств в КОМК лежит Управлении информационных технологий (далее УИТ).

Деятельность УИТ направлена на поддержание в рабочем состоянии и развитие программно-технического комплекса, установленного в Колледже.

Общее руководство осуществляется директором КОМК, непосредственное - начальником УИТ, который назначается на должность и освобождается приказом директором КОМК. Начальник УИТ организует и руководит работой специалистов УИТ.

В соответствии с «Положением об Управлении информационных технологий» на УИТ возложены следующие обязанности:

* Разработка основных направлений развития технической и информационной политики и технической базы в Колледже.
* Обеспечение бесперебойной работы автоматизированных рабочих мест, программных средств, сетевого оборудования, офисной АТС, каналов связи и доставки информации.
* Обеспечение целостности информационной среды КОМК.

В соответствии с обязанностями, лежащими на УИТ, были выделены Отдел разработки и сопровождения программного обеспечения (ОРиСПО) и Отдел сопровождения системной инфраструктуры (ОССИ).

В соответствии с «Положением об отделе разработки и сопровождения ПО» ОРиСПО:

* Осуществляет разработку технических проектов ПО, построение архитектуры программных систем, определение технических и программных требований к ПО.
* Проводит анализ поступающей информации о сбоях в работе ПО, связанных с ошибками в ПО и принимает меры к их оперативному устранению.
* Оказывает консультативную помощь структурным подразделениям по вопросам функционирования используемого программного обеспечения и используемым технологиям.
* Согласно получаемой от структурных подразделений информации, осуществляет доработку и разработку нового ПО.
* Осуществляет разработку новых версий ПО.
* Осуществляет предварительное тестирование новых версий ПО на правильность реализации алгоритмов и соответствие техническому заданию.
* Осуществляет передачу структурным подразделениям новых версий ПО.
* Осуществляет разработку документации по установке ПО и руководства администратора.
* Осуществляется изучение и внедрение новых информационных технологий в части разработки ПО и его совершенствования.

Возложение на отдел функций, не предусмотренных настоящим Положением, не допускается.

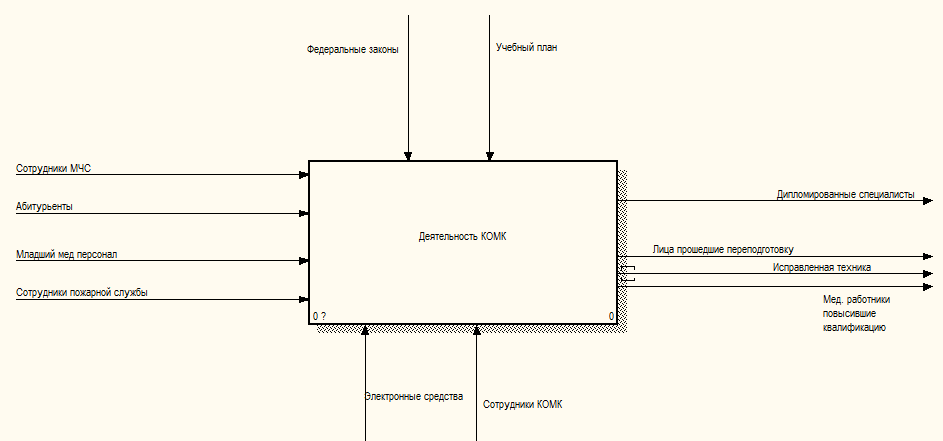
В соответствии с «Положением об отделе сопровождения системной инфраструктуры» ОССИ:

* Проводит работы по установке и сопровождению программных продуктов, обеспечивающих функционирование Колледжа.
* Обеспечивает внедрение и сопровождение Интранет-Интернет технологий в Колледже и его подразделениях.
* Администрирует WWW, E-mail, DNS, Active Directory, WSUS, Samba, ftp - сервера, межсетевые экраны, роутеры, коммутаторы, сервера общего пользования.
* Организует по поручению руководства проведение экспертизы информационных сетей.
* Контролирует процесс создания и использования в Колледже и его подразделениях информационных, автоматизированных и информационно-вычислительных систем и сетей.
* Администрирует доступ к различным сервисам СИ (подключением/отключение пользователей, программ, разграничение прав).
* Готовит, в пределах своей компетенции, для внесения в установленном порядке предложения по объемам и порядку финансирования работ в сфере сопровождения информационных технологий.
* Отслеживает новости и тенденции развития информационных систем, программных средств, баз и банков данных, взаимодействует со структурами предлагающими новые программные продуты на российском и международном рынках.
* Проводит обучение и консультации сотрудников Колледжа (в том числе дополнительных офисов и филиалов) по вопросам, относящимся к компетенции Отдела.
* Выполняет регламентные работы, связанные с обслуживанием Колледжа.
* Оборудует рабочие места пользователей компьютерной техникой
* Проводит монтажные работы ЛВС для функционирования Колледжа

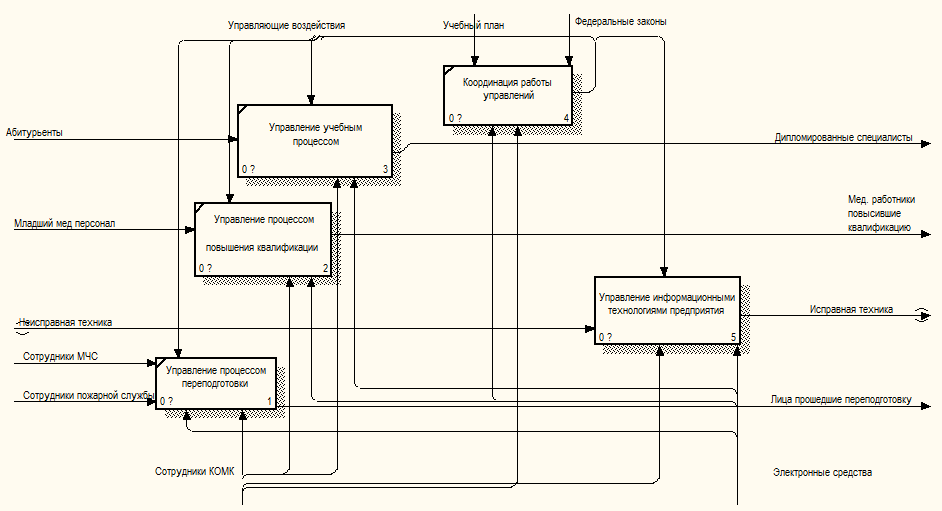
#### Общий перечень бизнес-процессов организации и их краткая характеристика, описание.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Код бизнес-процесса БП | Наименование бизнес-процесса БП | Степень важности процесса | Отдел владелец БП | Отдел исполнитель БП | БП – Источник информации | БП – Приемник выходной информации |
|  | Основные |  |  |  |  |  |
| КРУ | Координация работы управлений | А | Д | Д | - | УУП, УПП, УППК |
| УУП | Управление учебным процессом | А | УИМР | УИМР | КРУ | - |
| УПП | Управление процессом переподготовки | А | УДО | УДО | КРУ | - |
| УППК | Управление процессом повышения квалификации | А | УДО | УДО | КРУ | - |
| УИТП | Управление информационными технологиями предприятия | В | УИТ | УИТ | КРУ | ИТО, ИО |
|  | Обеспечивающие |  |  |  |  |  |
| ИТО | Инженерно-техническое обеспечение | В | ОССИ | ОССИ | УИТ | SD |
| ИО | Информационное обеспечение | В | ОРиСПО | ОРиСПО | УИТ | - |
| **SD** | **Сбор информации об тех. неисправностях и их исправление** | В | ОССИ | ОССИ | все БП | все БП |

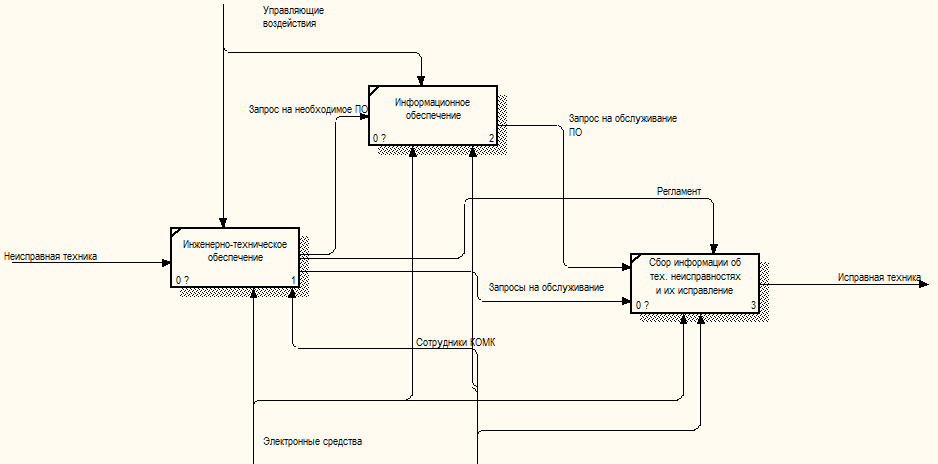
#### Контекстная диаграмма в нотации IDEF0 для всей организации и ее декомпозиция на составляющие бизнес-процессы (до уровня базового БП).



**Рисунок 1.2 Контекстная диаграмма**



**Рисунок 1.3 Декомпозиция контекстной диаграммы**



**Рисунок 1.4 Декомпозиция деятельности «Управление технологиями предприятия»**

#### Описание средства реализации процессов обмена между подразделениями и с внешними объектами.

Взаимодействие с внешними организациям происходит средствами:

* Бумажные носители;
* Средствами электронной почты;
* Государственные порталы (СУФД, Сбербанк АСТ, Парус, Госуслуги, ГИВЦ, РТС-Тендер и др.);
* ПО «Документооборот КОМК» 0.9.2.5. (частично реализует обмен документами между подразделениями).

Виды передаваемой информации:

* Отчеты в надзорные органы (Роспотребнадзор, Рособранадзор, ДОЗН, Роскомнадзор и др.)
* Информация о приемной компании (ФИС ГИА).
* Закупочная документация (Торговые площадки).
* Зарплатный пакет (Банки).

#### Описание схемы управления предприятием, бизнес-процесс управления на уровне предприятия, его организационную, методологическую, программную и аппаратную реализацию.

Управление КОМК осуществляется под руководством Директора. От него поступают указания заместителям по деятельностям КОМК (прим. Зам. директора по Учебной Работе). Они в свою очередь координируют работу подчиненных им отделений. Полная схема управления предприятием представлена в приложении 3. В пределах предприятия частично автоматизирован документооборот. Данную функцию выполняет ПО «Документооборот КОМК» 0.9.2.5.

#### Краткая характеристика базового бизнес-процесса и обоснование его выбора.

В качестве автоматизируемого бизнес-процесса был выбран процесс формирования меню и организации поставок. Данный бизнес-процесс был выбран по причине наличия большого количества тех. оборудования на предприятии, которое требует своевременного обслуживания специалистом, а также отсутствие ПО предназначенного для данной деятельности на предприятии как такового.

### Обследование функций, структуры и деятельности базового подразделения.

#### Название подразделения, цели подразделения, выполняемые функции, результаты деятельности подразделения

Базовым подразделением, на котором лежит обеспечение выбранного для автоматизации бизнес-процесса, является «Отдел сопровождения системной инфраструктуры».

Основными задачами Отдела являются:

* + Сопровождение и развитие сетевой и серверной подсистем Колледжа для обеспечения бесперебойного функционирования его служб и подразделений;
  + Обеспечение информационной безопасности серверной и сетевой компонент системной инфраструктуры (СИ) Колледжа, а также рабочих мест сотрудников отделов и подразделений Колледжа;
  + Организация связи, в частности: Доступа в сеть Интернет, ip – телефонии и видеосвязи.

Для решения поставленных задач Отдел осуществляет следующие функции:

* Проводит работы по установке и сопровождению программных продуктов, обеспечивающих функционирование Колледжа.
* Обеспечивает внедрение и сопровождение Интранет-Интернет технологий в Колледже и его подразделениях.
* Администрирует WWW, E-mail, DNS, Active Directory, WSUS, Samba, ftp - сервера, межсетевые экраны, роутеры, коммутаторы, сервера общего пользования.
* Организует по поручению руководства проведение экспертизы информационных сетей.
* Контролирует процесс создания и использования в Колледже и его подразделениях информационных, автоматизированных и информационно-вычислительных систем и сетей.
* Администрирует доступ к различным сервисам СИ (подключением/отключение пользователей, программ, разграничение прав).
* Готовит, в пределах своей компетенции, для внесения в установленном порядке предложения по объемам и порядку финансирования работ в сфере сопровождения информационных технологий.
* Отслеживает новости и тенденции развития информационных систем, программных средств, баз и банков данных, взаимодействует со структурами предлагающими новые программные продуты на российском и международном рынках.
* Проводит обучение и консультации сотрудников Колледжа (в том числе дополнительных офисов и филиалов) по вопросам, относящимся к компетенции Отдела.
* Выполняет регламентные работы, связанные с обслуживанием Колледжа.
* Оборудует рабочие места пользователей компьютерной техникой
* Проводит монтажные работы ЛВС для функционирования Колледжа

#### Общедоступные средства ввода-вывода, отображения, обмена информацией.

Обмен данными между подразделениями, в частности документооборот частично автоматизированы ПО «Документооборот КОМК» 0.9.2.5. Ручная передача документов так же все еще в ходу.

#### Описание процесса обмена информацией между рабочими местами и с внешними объектами.

Процесс обмена информации между местами происходит за счет локальной сети, которая подключена ко всем рабочим станциям, так же обмен информацией происходит через БД, которая расположена на главном сервере КОМК.

На серверах КОМК осуществлены следующие средства защиты:

На Windows серверах установлен и настроен Kaspersky endpoint security 10, на linux – настроен iptables в качестве брандмауэра. На пограничном маршрутизаторе mikrotik также настроен брандмауэр. Присутствует разграничение пользователей по ролям при помощи Active directory, а так же разграничение трафика по vlan.

#### Описание штатного расписания сотрудников и выполняемых ими функций.

В соответствии с должностной инструкцией специалиста отдела сопровождения системной инфраструктуры специалист отдела сопровождения системной инфраструктуры:

* + Организует проектирование и модернизацию компьютерной и офисной оргтехники.
  + Обеспечивает потребности подразделений колледжа расходными материалами.
  + Обеспечивает установку новых рабочих мест сотрудникам колледжа.
  + Обеспечивает программно-техническое взаимодействие с разработчиками, поставщиками техники и программного обеспечения.
  + Осуществляет взаимодействие с другими подразделениями колледжа.
  + Планирует, организует учебный процесс для работников отдела.
  + Вносит предложения вышестоящему руководителю по подбору и расстановке кадров.
  + Создает необходимые условия для продуктивного труда, принимает меры к повышению уровня и эффективности работы отдела, повышению профессионального уровня работников.
  + Консультирует сотрудников колледжа по всем возникающим производственным вопросам и по вопросам технической эксплуатации оргтехники колледжа.
  + Соблюдает внутренний распорядок отдела.
  + Постоянно повышает свой профессиональный уровень.
  + Исполняет распоряжения и приказы директора колледжа, руководителя отдела сопровождения системной инфраструктуры, руководителя управления ИТ.
  + Информирует руководство об имеющихся недостатках в работе предприятия, принимаемых мерах по их ликвидации.

Режим работы специалиста определяется в соответствии с Правилами внутреннего трудового распорядка, установленными на предприятии.

Работает в режиме нормированного рабочего дня с 8.00 до 17.00, с понедельника по пятницу.

* ***Документы, регламентирующие деятельность;***

Деятельность подразделения регламентируют следующие документы:

* Положение об отделе сопровождения системной инфраструктуры;
* Должностная инструкция начальника отдела сопровождения системной инфраструктуры;
* Должностная инструкция специалиста отдела сопровождения системной инфраструктуры.
* ***Входящие документы, исходящие документы в виде схемы документооборота и табеля документооборота.***

Схема документооборота подразделения

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Номер документа | Наименование документа | Откуда приходит / исходит документ | Куда уходит | Операции, выполняемые над документом | Ответственный за выполнение операций над документом |
| СЗ | Служебная записка | Источник любой | УИТ | Формирование записки | Сотрудник ОССИ |
| АПОС | Акт о перемещении ОС | Начальник ОССИ | Б (Бухгалтерия) | Составление акта | Глав. бух. |

Табель документооборота

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Номер документа | Наименование документа | Тип документа | Частота документа за временной период (раз/месяц, раз/раб. день) | Ответственный за документ |
| СЗ | Служебная записка | Внутренний, исходящий | > 2 раз/раб.день | Сотрудник ОССИ |
| АПОС | Акт о перемещении ОС | Внутренний, исходящий | 5 раз/месяц | Глав. бух. |

* ***Перечень имеющихся организационных документов.***
* *Положение об отделе разработки и сопровождения программного обеспечения;*
* *Положение об отделе сопровождения системной инфраструктуры;*
* *Положение об Управлении информационных технологий.*

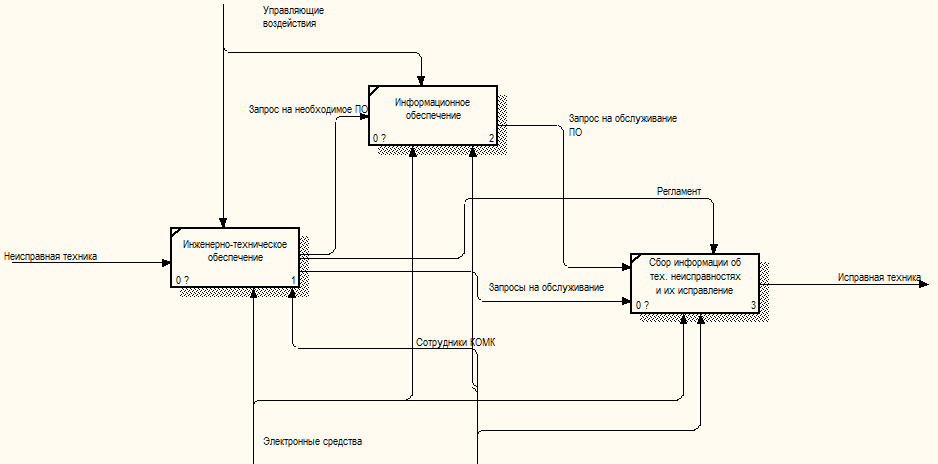
### Выявление бизнес-процессов базового подразделения.

#### Перечень и описание бизнес-процессов, выполняемых подразделением, представленные в табличном виде.

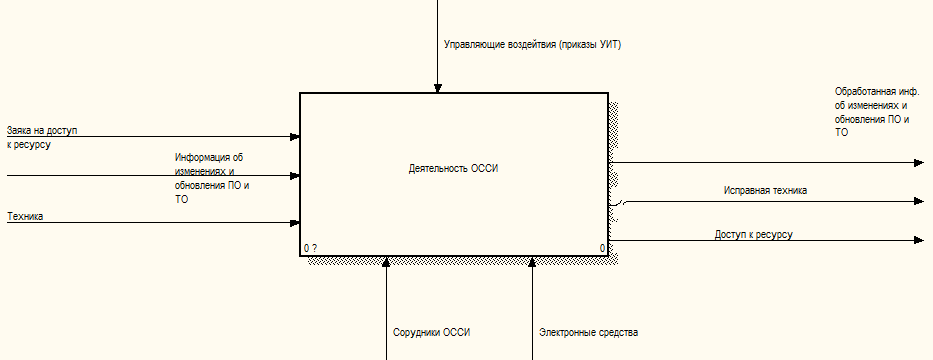
*ОССИ обеспечивает следующие бизнес-процессы:*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Код бизнес-процесса БП | Наименование бизнес-процесса БП | Степень важности процесса | БП – Источник информации | БП – Приемник выходной информации |
| **SD** | **Сбор информации об тех. неисправностях и их исправление** | **A** | ИТО, ОНИС | АДСИ |
| АДСИ | Администрирование доступа к различным сервисам СИ | B | SD | Все БП |
| ОНИС | Отслеживание новостей и тенденций развития ИС | В | - | SD |
| ИТО | Инженерно-техническое обеспечение | А | - | SD |

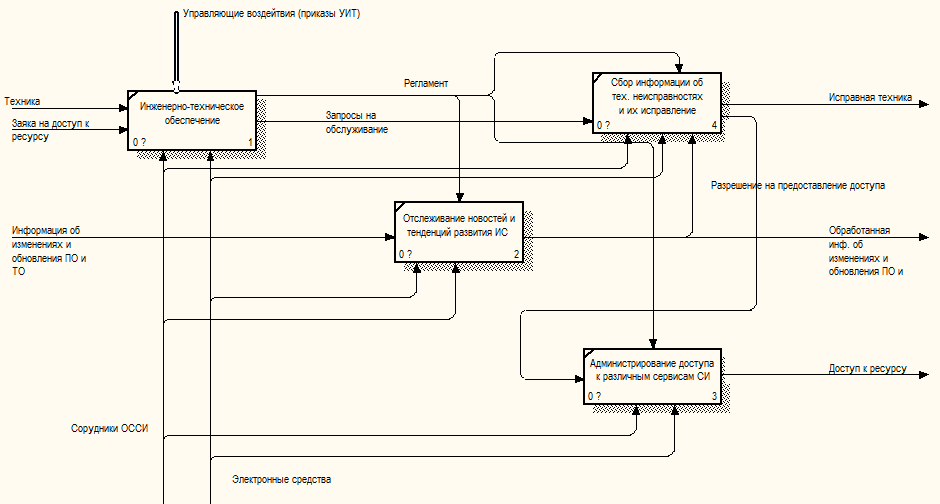
#### Состав и взаимосвязь бизнес-процессов базового подразделения, представленная в виде функциональной диаграммы в нотации IDEF0.



**Рисунок 1.5 Положение базового БП в деятельности КОМК**



**Рисунок 1.6 Контекстная диаграмма деятельности ОССИ**



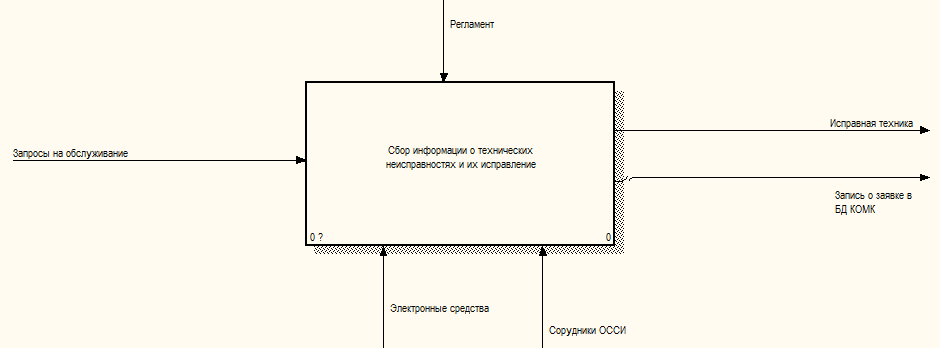
**Рисунок 1.7 Декомпозиция контекстной диаграммы**

#### Подробное описание и анализ базового БП в табличном виде.

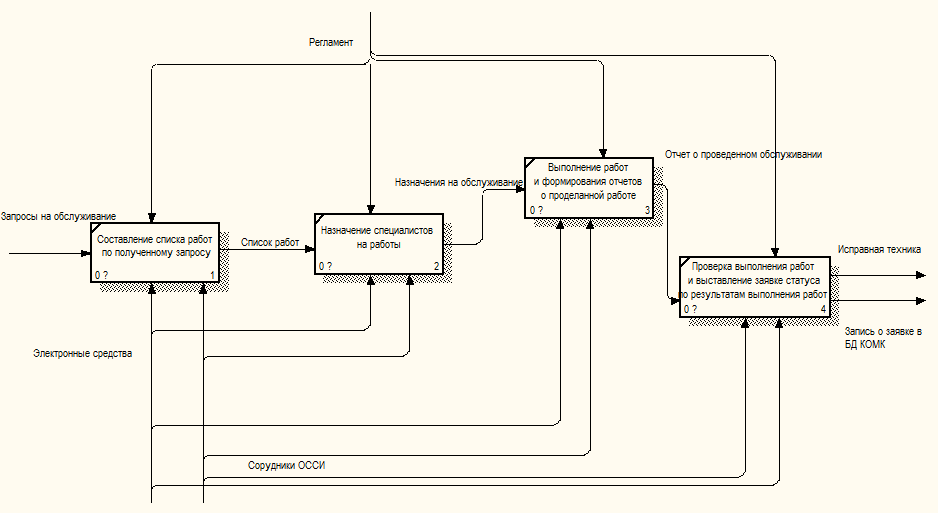
Далее выполняется подробное описание базового бизнес-процесса. Для этого необходимо привести следующие данные:

|  |  |
| --- | --- |
| Необходимые данные | Значение |
| Условия начала выполнения бизнес-процесса | Поступление заявки от любого сотрудника КОМК на решение проблемы связанной с ТО. |
| Документы и данные, необходимые для выполнения бизнес-процесса, их источники. | Необходимые документы и данные определяются индивидуально для каждой заявки, так, например, для добавления сотруднику доступа к различным сервисам СИ необходимо разрешение на добавление от отдела ИТО |
| Документы, создаваемые в результате выполнения бизнес-процесса, и их получатели. | В результате выполнения БП формируется запись в БД КОМК о выполненной работе. Возможен доступ к просмотру состояния собственных заявок для рядового пользователя, а так же всех отправленных в отдел заявок для сотрудников ОССИ. |
| Действующие лица, принимающие участие в выполнении бизнес-процесса. | Действующих лиц можно разделить на 2 группы:   * Сотрудники КОМК (пользователи) * Сотрудники ОССИ (специалисты тех. поддержки и исполнители) |
| Материальные ценности, необходимые для выполнения бизнес-процесса, если таковые есть. | В зависимости от возникших неполадок, могут как присутствовать, так и отсутствовать (Например: Не печатает принтер, в данном случае в зависимости от причины неполадок может потребоваться например краска, бумага, возможно замена принтера либо вообще никаких расходников не потребуется). |
| Материальные ценности – результат выполнения бизнес-процесса, если таковые есть. | - |
| Результаты выполнения бизнес-процесса (кроме материальных ценностей). | Результатом выполнения БП считается исправление проблем и поддержание общей работоспособности ТО. |
| Цель данного бизнес-процесса, его место и роль в общих задачах (процессах) компании. | Поддержание ТО в рабочем состоянии, как следствие отсутствие каких-либо задержек в работе других отделов, а так же увеличение срока службы ТО. |
| Проблемы, возникающие при выполнении бизнес-процесса. | Неспособность пользователя объяснить суть проблемы.  Неспособность специалиста ОССИ выявить источник проблемы. |
| Нештатное завершение (выполнение) бизнес-процесса. | Таковых нет. В случае если заявка пользователя не может быть выполнена, в соответствующей графе записи о статусе заявки это указывается. |
| Последовательность действий выполнения бизнес-процесса. | * *Получение заявки от пользователя* * *Создание специалистом списка необходимых работ для данной заявки* * *Выполнение исполнителями всех работ по полученному списку* * *Проверка выполнения всех работ и выставление заявке статуса, о завершении либо приостановке работы над ней.* |

#### IDEF0 диаграмма для базового БП.



**Рисунок 1.8 Контекстная диаграмма базового процесса**



**Рисунок 1.9 Декомпозиция базового процесса на деятельности**

### Анализ существующей реализации базового бизнес-процесса.

В качестве базового (автоматизируемого) процесса был выбран процесс «Сбора информации о технических неисправностях и их исправление». В КОМК отсутствует автоматизация данного процесса.

Сам базовый процесс предполагает следующие этапы:

1. Оформление пользователем заявки на тех. обслуживание. В ней пользователь оставляет свои контактные данные и кратко описывает проблему, после этого заявка отправляется в ОССИ.
2. По поступлении в отдел, полученная заявка обрабатывается специалистом. Специалист формирует список работ, требуемых для выполнения тех. обслуживания.
3. После формирования списка работ, на каждую из выделенных работ назначается работник ОССИ.
4. По окончании каждой из работ, работник ОССИ составляет отчет о проделанной им работе.
5. По выполнении всего списка работ, специалист выставляет заявке статус, соответствующий результату тех. обслуживания.

Базовый процесс можно разложить на следующие подпроцессы:

1. Составление списка работ по полученному запросу;
2. Назначение специалистов на работы;
3. Выполнение работ и оформление отчетов о проделанной работе;
4. Проверка выполнения работ и выставление заявке статуса по результатам выполнения работ.

## Формулировка требований к системе

Определение деятельностей, требующих автоматизации, кандидатов в требования.

#### Состав бизнес-процессов предприятия.

Состав бизнес-процессов приведен на рисунке 1.10.

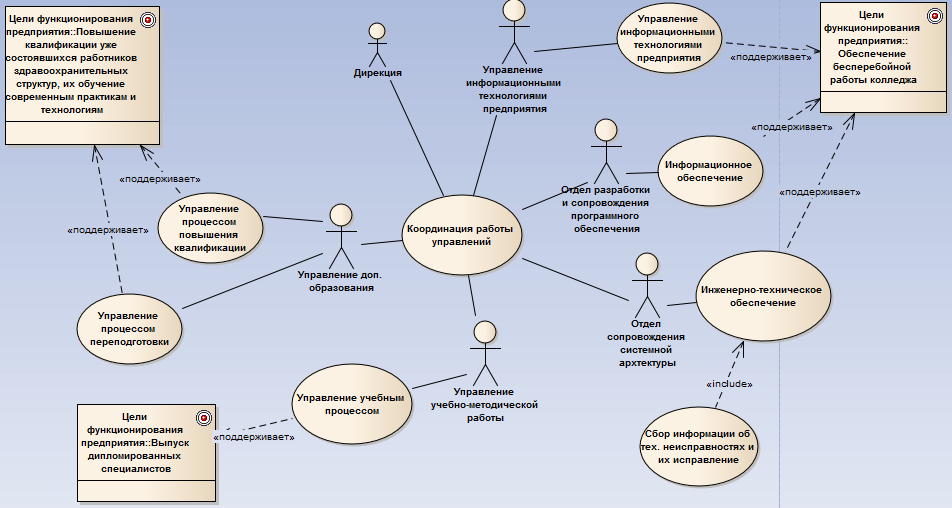


Рисунок 1.10. Состав бизнес-процессов

* Дирекция - осуществляет координацию работы управлений;
* Управление информационными технологиями предприятия - разработка основных направлений развития технической и информационной политики и технической базы в Колледже; Обеспечение бесперебойной работы автоматизированных рабочих мест, программных средств, сетевого оборудования, офисной АТС, каналов связи и доставки информации;
* Отдел разработки и сопровождения программного обеспечения - осуществляет информационное обеспечение;
* Отдел сопровождения системной архитектуры осуществляет инженерно-техническое обеспечение.
  + Сбор информации об тех. неисправностях и их исправлении (система Service desk)
* Отдел управление доп образованием осуществляет управление процессом переподготовки и процессом повышения квалификации
* Отдел управление учебно-методической работой осуществляет управление учебным процессом

#### Содержание бизнес-процессов.

Для подробного анализа выбран следующий процесс: “Сбор информации об тех. неисправностях и их исправление”.

Содержание бизнес-процесса “Сбор информации об тех. неисправностях и их исправление” приведено на рисунке 1.11.

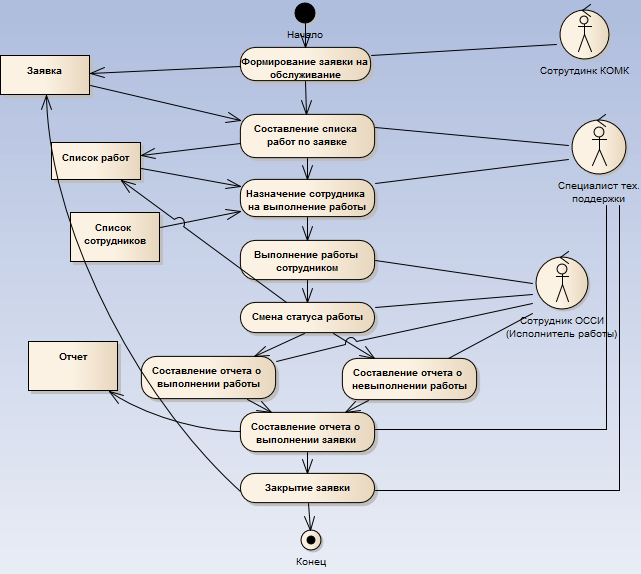


Рисунок 1.11. Сбор информации об тех. неисправностях и их исправление

Работы выполняемые сотрудником КОМК:

* Формирование заявки на обслуживание – сотрудник пишет заявку по поводу неисправности оборудования специалисту тех. поддержки;

Работы выполняемые специалистом тех поддержки:

* Составление списка работ по заявке;
* Назначение сотрудника на выполнение работы;
* Составление отчета о выполнении заявки;
* Закрытие заявки;

Работы выполняемые сотрудником ОССИ:

* Выполнение работы сотрудником;
* Смена статуса работы;
* Составление отчета по выполнению/невыполнению работы;
* Отчет о проверке направляется в соответствующие органы/отделы.

Описание объектов:

* Заявка – является выходным объектом формирования заявки пользователем; входным объектом для составления списка работ;
* Список работ – является выходным объектом для составления списка работ; входным объектом для назначения сотрудника на работу;
* Список сотрудников– является входным объектом для назначения сотрудника на работу;
* Отчет – выходным для закрытия заявки.

#### Классы-объекты предметной области.

Описание классов-объектов предметной области приведено на рисунке 1.12.

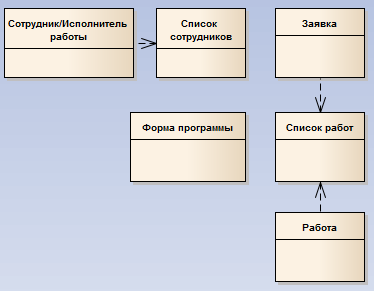


Рисунок 1.12. Описание классов-объектов предметной области

Выявленные классы:

* Работа – информационный объект, содержащий описание необходимых действий;
* Список работ – информационный объект, представляет собой список всех необходимых для заявки работ;
* Заявка – информационный объект, представляет собой запрос на обслуживание;
* Сотрудник/Исполнитель работы – информационный объект, представляет собой данные о сотруднике;
* Список сотрудников - информационный объект, представляет собой список всех исполнителе;
* Форма программы – информационный объект, представляет собой интерфейс программы.

#### Автоматизируемые элементы.

Предполагаемые автоматизируемые элементы приведены на рисунке 1.13.

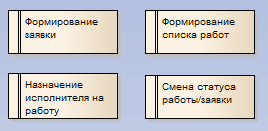


Рисунок 1.13. Автоматизируемые элементы бизнес-процессов

Формирование заявки соответствует функции «Формирование заявки». Автоматизация данного процесса позволит сократить время составления заявки и уменьшить количество ошибок.

Формирование списка работ соответствует функции «Составление работ». Автоматизация данного процесса позволит быстро составить план выполнения заявки.

Назначение исполнителей на работу «Управление исполнением необходимых работ». Автоматизация данного процесса позволит обеспечить более быстрый и точный подбор кадров для выполнения задачи.

Смена статуса работы/заявки соответствует функции «Изменение статуса работ». Автоматизация данного процесса позволит быстро получить информацию по статусу работы.

Определение состава функциональных требований к системе.

Соответствие функциональных требований автоматизируемым бизнес-решениям отобразим в виде матрицы трассировки (таблица 1).

Таблица 1. Требования

|  |  |  |
| --- | --- | --- |
| **№** | **Бизнес-процесс** | **Требование** |
| 1 | Формирование заявки | Обеспечение возможности создания заявки в фиксированной форме. |
| 2 | Формирование списка работ | Обеспечение возможности формирования списка работ на основе поступившей заявки;  Обеспечение формирования типов работ. |
| 3 | Назначение исполнителя на работу | Обеспечение возможности назначения на работу сотрудника;  Обеспечение возможности разграничения прав пользователей и формирования их списка умений. |
| 4 | Смена статуса работы/заявки | Обеспечение возможности смены статуса заявки с сохранением истории ее исполнения. |
| Обеспечение возможности смены статуса работы с сохранением истории ее исполнения. |

Диаграмма соответствия функциональных требований автоматизируемым бизнес-решениям приведена на рисунке 1.14.

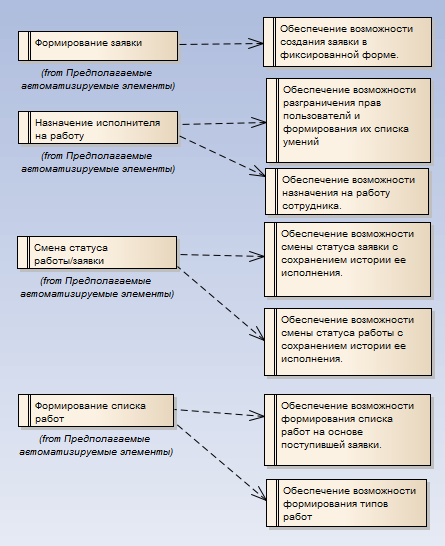


Рисунок 1.14. Диаграмма соответствия функциональных требований автоматизируемым бизнес-решениям

#### Состав подсистем.

Для отображения выявленных подсистем построим матрицу трассировки «подсистема – функциональные требования» (таблица 2).

Таблица 2. Подсистемы

|  |  |
| --- | --- |
| **Подсистема** | **Требование** |
| Подсистема работы с заявками | Обеспечение возможности формирования списка работ на основе поступившей заявки. |
| Обеспечение возможности назначения на работу сотрудника. |
| Обеспечение возможности смены статуса заявки с сохранением истории ее исполнения. |
| Подсистема создания заявок | Обеспечение возможности создания  заявки в фиксированной форме. |
| Подсистема работы с работами по заявке | Обеспечение возможности смены статуса работы с сохранением истории ее исполнения. |
| Обеспечение возможности формирования типов работ. |
| Обеспечение возможности разграничения прав пользователей и формирования их списка умений |

Диаграмма выявленных подсистем приведена на рисунке 1.15.

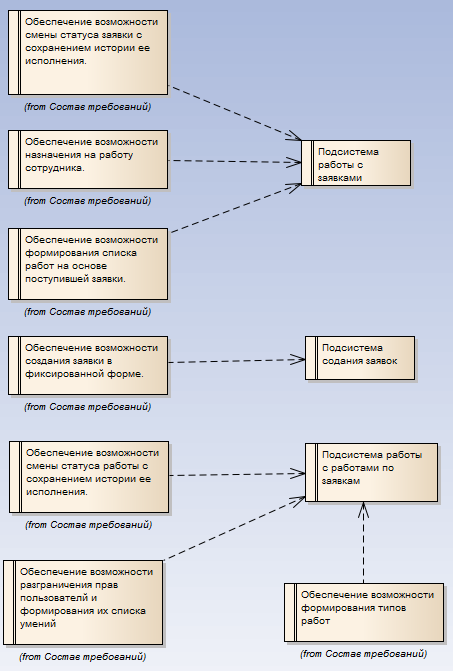


Рисунок 1.15. Диаграмма выявленных подсистем

#### Варианты использования.

Диаграмма вариантов использования, отражающая функциональные требования к системе и сценарии, их реализующие, приведены на рисунке 1.16.

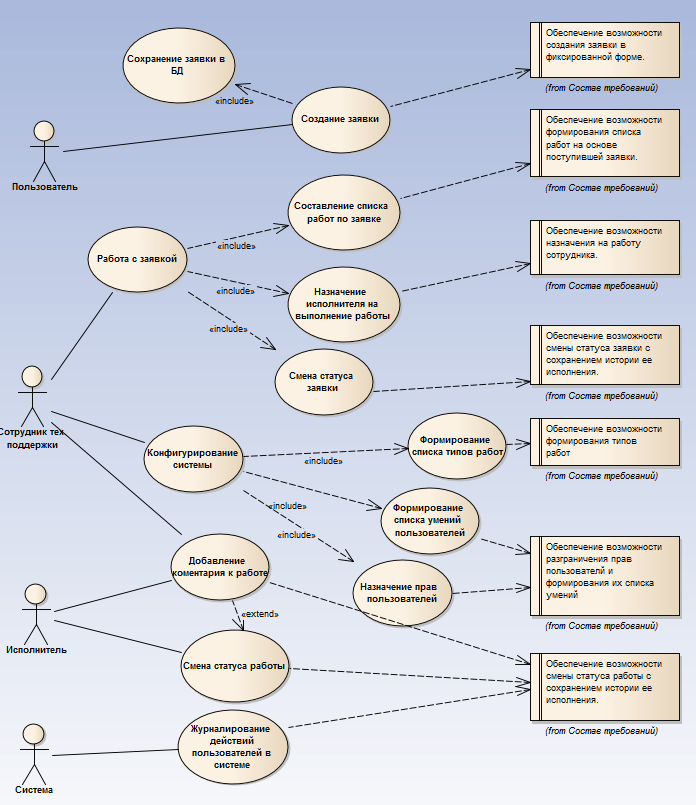


Рисунок 1.16. Диаграмма вариантов использования

Функции, выполняемые пользователем:

* Создание заявки, включает в себя сохранение заявки в БД.

Функции, выполняемые исполнителем:

* Смена статуса работы, которая расширяется возможность добавления комментария.

Функции, выполняемые сотрудником тех. поддержки:

* Работа с заявкой, которая включает в себя: составление списка работ по заявке, назначение исполнителя на выполнение работы и смена статуса заявки;
* Конфигурирование системы, которая включает в себя: формирование списка типов работ, формирование списка умений пользователей, назначение прав пользователей;
* Добавление комментария к работе.

Функции, выполняемые системой:

* Журналирование действий пользователей в системе.

### Разработка сценариев, реализующих требования к системе.

#### Содержание сценариев.

Проведем анализ следующих сценариев: «Создание заявки», «Работа со справочником», «Составление отчета», «Авторизация пользователя», «Журналирование действий пользователя в системе».

Сценарии выполнения остальных ВИ рассмотрены не будут, т.к. они имеют косвенное отношение к работе с заявками пользователей.

Диаграмма деятельности сценария «Создание заявки» приведена на рисунке 1.17.

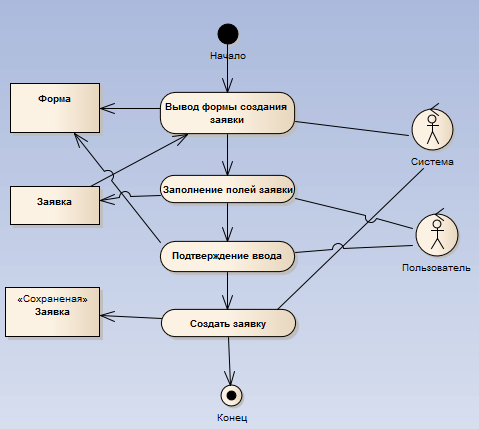


Рисунок 1.17. Создание заявки

В результате работы процесса выявлены следующие объекты:

* Граничные объекты:
* Форма – содержит формы «Форма создания заявки». Позволяет создавать, изменять, удалять, искать и выгружать документ в Word.
* Сущности:
* Заявка – содержит шаблон для создания заявки.
* Сохраненная заявка – содержит описание возникшей проблемы с указанием даты подачи заявки и заявителя

Сценарии выполняются пользователем и системой:

* Вывод формы заявки, открывается форма создания заявки с полем текста заявки и значениями текущей даты и описания заявителя (выполняется системой);
* Заполнение полей заявки (выполняется пользователем);
* Подтверждение ввода (выполняется пользователем);
* Сохранение заявки в БД (выполняется системой).

Диаграмма деятельности сценария «Работа с заявкой» приведена на рисунке 1.18.

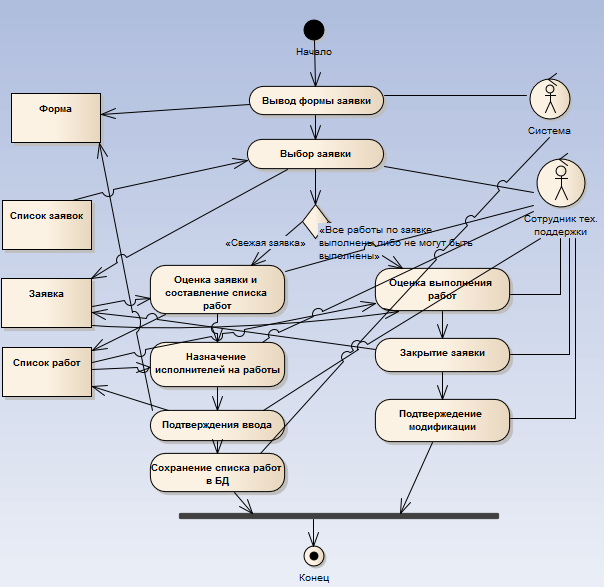


Рисунок 1.18. Работа с заявкой

В результате работы процесса выявлены следующие объекты:

* Граничные объекты:
* Форма – содержит формы «Добавить работу», «Работа с заявкой». Позволяет назначать исполнителям работы по заявке.
* Сущности:
* Список заявок – содержит заявки пользователей;
* Заявка – содержит данные о конкретной заявке;
* Список работ - содержит данные о работах по заявке.

В выполнении сценариев участвуют система и сотрудник тех. поддержки:

* Выводится форма работы с заявками;
* Сотрудник выбирает заявку:
  + Закрытие заявки;
    - Выполняется оценка выполнения работ назначенных на заявку;
    - В случае если все работы выполнены, либо не могут быть выполнены заявка закрывается;
    - Подтверждается закрытие заявки, данные о закрытии записываются в БД предприятия.
  + Обработка заявки;
    - Сотрудник просматривает заявку и составляет к ней список работ;
    - На назначенные работы выбирается исполнитель;
    - Если обработка заявки окончена, сотрудник подтверждает окончание работы с заявкой;
    - Данные о заявке и назначенных на нее работах сохраняются в БД.

Диаграмма деятельности сценария «Смена статуса работы» приведена на рисунке 1.19.

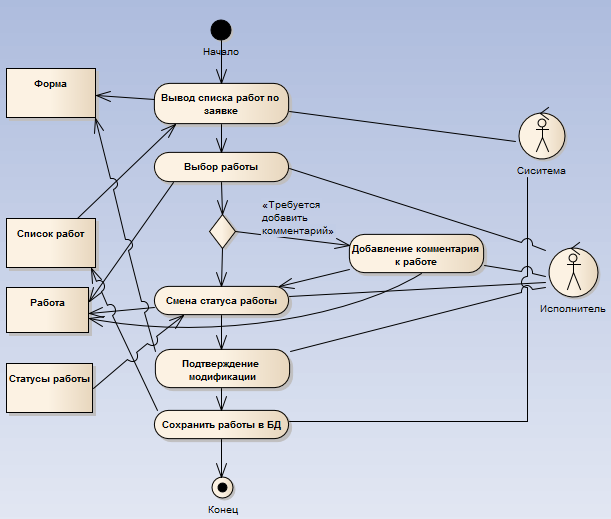


Рисунок 1.19. Смена статуса работы

В результате работы процесса выявлены следующие объекты:

* Граничные объекты:
* Форма – содержит формы «Смена статуса работы». Позволяет исполнителю работы изменять статус работы, а также комментировать её.
* Сущности:
* Заявка - содержит описание заявки, ее статус, а также список работ, требуемых для ее закрытия.
* Работа - содержит описание работы.
* Список работ - содержит все работы, требуемые к выполнению.
* Статусы работы- содержит все статусы возможное для работы.

В выполнении сценария участвуют система и исполнитель:

* Вывод списка работ по заявке (выполняется системой);
* Выбор работы (выполняется исполнителем);
  + Если исполнитель считает нужным, выполняется добавление комментария к работе (выполняется исполнителем);
* Подтверждение модификации (выполняется исполнителем):
* Обновление БД, изменения статуса работы по заявке фиксируется в БД (выполняется системой).

Диаграмма деятельности сценария «Авторизация пользователя» приведена на рисунке 1.20.

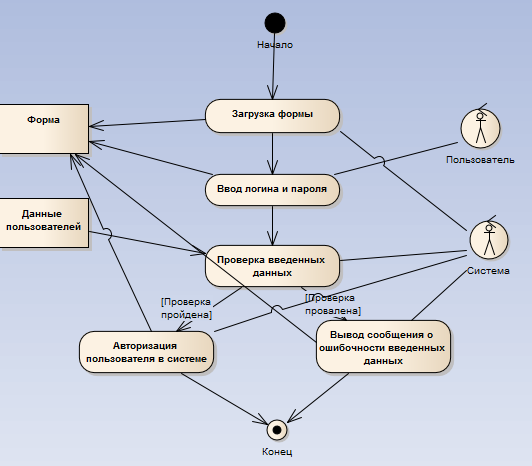


Рисунок 1.20. Авторизация пользователя

В результате работы процесса выявлены следующие объекты:

* Граничные объекты:
* Форма – содержит форму «Авторизация пользователя». Позволяет пользователю авторизоваться за свой аккаунт.
* Сущности:
* Данные пользователя - содержит данные о пользователях зарегистрированных в системе.

В выполнении сценария участвуют система и исполнитель:

* Вывод формы авторизации (выполняется системой);
* Ввод данных (выполняется исполнителем);
* Проверка данных (выполняется системой):
  + Если проверка пройдена: Авторизация пользователя в системе (выполняется системой);
  + Если проверка провалена: Вывод сообщения об ошибочности введенных данных (выполняется системой).

Диаграмма деятельности сценария «Журналирование действий пользователя в системе» приведена на рисунке 1.21.

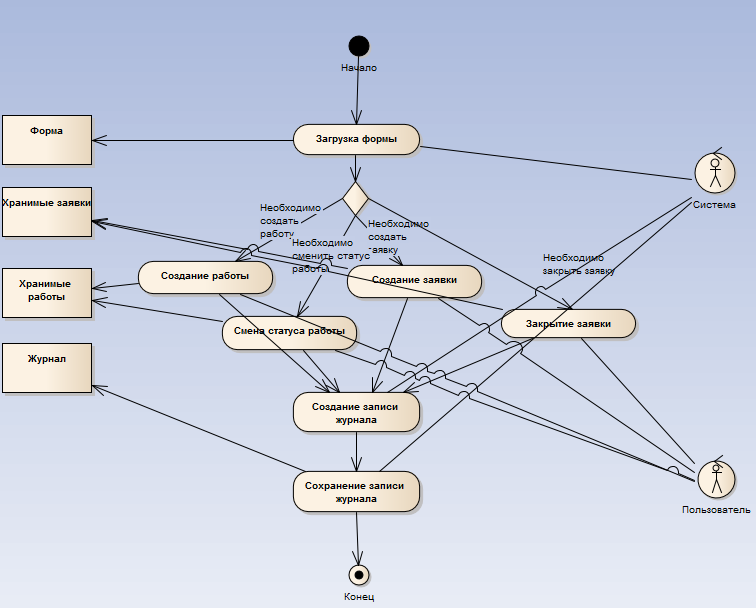


Рисунок 1.21. Журналирование действий пользователя в системе

В результате работы процесса выявлены следующие объекты:

* Граничные объекты:
* Форма – содержит формы «Смена статуса работы». Позволяет исполнителю работы изменять статус работы, а также комментировать её.
* Сущности:
* Хранимые заявки - содержит хранимые заявки;
* Хранимые работы - содержит хранимые работы;
* Журнал - содержит записи о действиях пользователей.

В выполнении сценария участвуют система и пользователь:

* Загрузка формы(выполняется системой);
  + Создание работы (выполняется пользователем);
  + Смена статуса работы (выполняется пользователем);
  + Создание заявки (выполняется пользователем);
  + Закрытие заявки (выполняется пользователем);
* Создание записи журнала (выполняется системой);
* Сохранение записи в журнал (выполняется системой).

Определение требований к интерфейсу.

Требования к интерфейсу:

● Возможность составления заявок;

● Возможность обработки заявок сотрудником тех. поддержки;

● Возможность добавления работ к заявке;

● Возможность смены статуса работы;

● Возможность добавления комментария к работе;

● Возможность смены статуса заявки;

● Возможность назначения исполнителей на работы.

Прототип пользовательского интерфейса в виде диаграммы экранных форм приведен на рисунке 1.22.

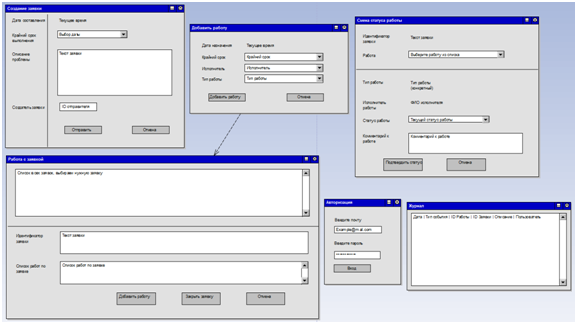
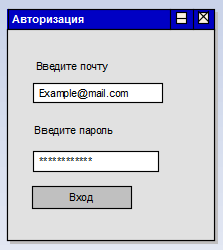


Рисунок 1.22. Интерфейс подсистемы «Внутренний документооборот ДОН Кемеровской области»

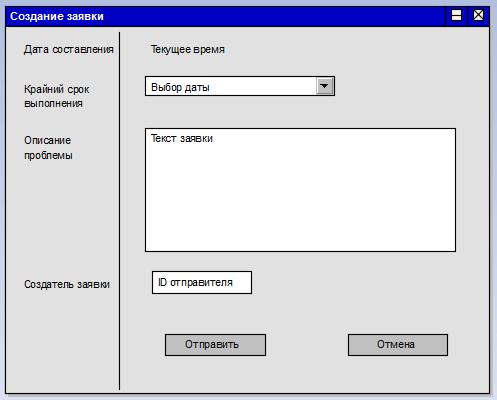
Состав форм:



**Рисунок 1.23 - Форма “Авторизация”**

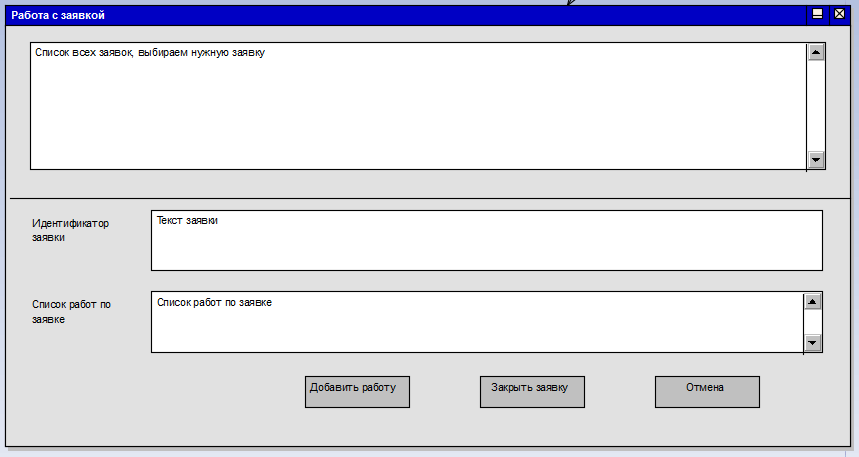
Форма “Авторизация”, предназначена для авторизации пользователей.

Содержит два поля для ввода логина (в конкретном случае использован адрес электронной почты пользователя) и пароля, а также кнопку подтверждения ввода.



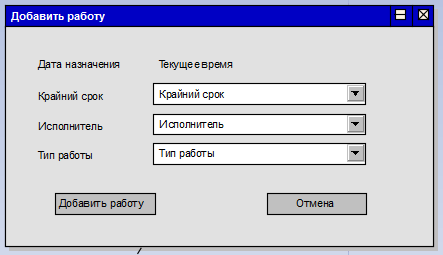
**Рисунок 1.24 - Форма “Создание заявки”**

Форма “Создание заявки”, предназначена для создания заявок пользователями. На форме необходимо выбрать крайний срок выполнения и ввести текст заявки, форма автоматически определяет пользователя, отправляющего заявку.



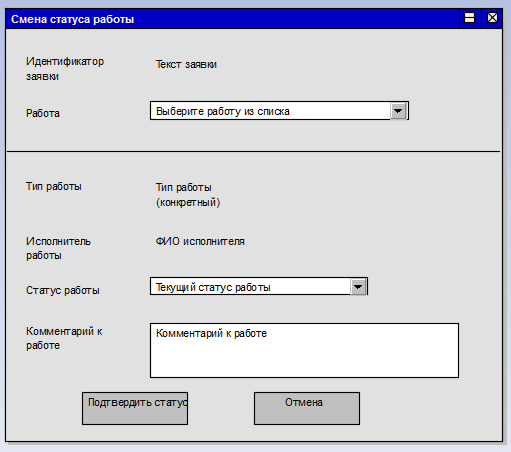
**Рисунок 1.25 - Форма “Работа с заявкой”**

Форма “Работа с заявкой”, предназначена для обработки заявок сотрудниками тех. поддержки. Предполагается выбор заявки из списка, затем либо осуществляется переход на форму “Добавить работу” для создания работ по заявке, либо закрытие заявки.



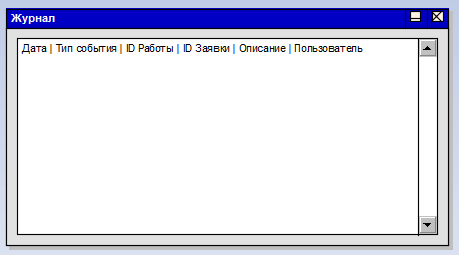
**Рисунок 1.26 - Форма “Добавить работу”**

Форма “Добавить работу”, предназначена для создания работы по заявке, для создания работы требуется назначить исполнителя работы и крайний срок ее выполнения, а также выбрать тип работы.



**Рисунок 1.27 - Форма “Смена статуса работы”**

Форма “Смена статуса работы”, предназначена для изменения статуса работы исполнителем, а также добавления комментариев к работе.



**Рисунок 1.28 - Форма “Журнал”**

Форма “Журнал”, предназначена для мониторинга действий пользователей в системе. В частности в данной форме регистрируется создание заявок/работ, смены статусов заявок/работ.

Итоговый состав форм может отличаться от указанного, но заявленная функциональность при этом не должна быть потеряна.

Окончательная формулировка требований к системе.

#### Функциональные требования к системе.

1. Обеспечение возможности создания заявки в фиксированной форме (кр. «Создание заявки»).

* Описание: В программе должна быть реализована возможность создания заявки пользователем. При этом должна быть обеспечена возможность задания пользователем текста заявки. Приложение при создании заявки должно автоматически добавлять к ней идентификатор пользователя, оставившего заявку, регистрировать дату создания заявки и выставлять соответствующий статус («Ожидает»);
* Статус требования: принято;
* Тестовый пример: Зарегистрировать заявку от пользователя Андреев Семён.
  + Исходные данные:
    - Андреев Семен авторизован в системе;
    - Текст заявки - “Клавиатура не реагирует на нажатия клавиш (лол, как он тогда заявку составил?)”;
    - Время создания заявки - 01.02.2017 10:36:00;
    - Крайний срок выполнения заявки - 02.02.2017;
    - Тех.содрудник которому отправляется заявка - Колыпатов Михаил.
  + Результаты:
    - В приложении создана заявка со следующими параметрами: Дата поступления - 01.02.2017, Последнее изменение - 01.02.2017 10:36:00, Выполнить до - 02.02.2017, Текст - “Клавиатура не реагирует на нажатия клавиш”, Отправитель - Андреев Семен, Обработал - Колыпатов Михаил, Статус заявки - Обрабатывается.
    - В БД создана соответствующая запись.

1. Обеспечение возможности формирования списка работ на основе поступившей заявки (кр. «Формирование списка работ»);
   * Описание: В программе должна быть реализована формирования списка работ по заявке. При этом должна быть обеспечена возможность выбора заявки, к которой относится работа, задания типа работы, крайнего срока ее выполнения. Приложение при создании работы должно автоматически добавлять к ней идентификатор специалиста обработавшего заявку, регистрировать дату создания работы и выставлять соответствующий статус («Ожидает»);
   * Статус требования: принято;
   * Тестовый пример: Обработать заявку от пользователя Андреев Семён от имени работника тех. поддержки Колыпатова Михаила. Составив список работ по поступившей заявке.
     + Исходные данные:
       - Колыпатов Михаил авторизован в системе;
       - Текст заявки - “Клавиатура не реагирует на нажатия клавиш”;
       - Время создания работы к заявке - 01.02.2017 10:40:00;
       - Крайний срок выполнения работы - 02.02.2017;
       - Тип работы - работа с ПК;
     + Результаты:
       - В приложении создана работа со следующими параметрами: Дата поступления - 01.02.2017, Последнее изменение - 01.02.2017 10:40:00, Выполнить до - 02.02.2017, Статус работы – ожидает выполнения, Тип работы - работа с ПК, Текст заявки - Клавиатура не реагирует на нажатия клавиш.
       - В БД создана соответствующая запись.
2. Обеспечение возможности назначения на работу исполнителя (кр. «Назначение исполнителя на работу»);
   * Описание: В программе должна быть реализована возможность назначения исполнителя на выполнение работы, а так же смены исполнителя на работе с созданием соответствующей записи в системе;
   * Статус требования: принято;
   * Тестовый пример: Назначить на работу «работа с ПК» по заявке «Клавиатура не реагирует на нажатия клавиш» в качестве исполнителя Смирнова Евгения от имени сотрудника тех. поддержки Колыпатова Михаила.
     + Исходные данные:
       - Колыпатов Михаил авторизован в системе;
       - Время назначения работы на исполнителя - 01.02.2017 10:50:00;
       - Крайний срок выполнения работы - 02.02.2017;
       - Тип работы - работа с ПК;
       - Работник, назначаемый на работу - Смирнов Евгений;
     + Результаты:
       - В приложении создана заявка со следующими параметрами: Дата поступления - 01.02.2017, Последнее изменение - 01.02.2017 10:50:00, Выполнить до - 02.02.2017, Выполняет - Смиирнов Евгений, Статус работы – ожидает выполнения, Тип работы - работа с ПК.
       - В БД изменена соответствующая запись.
3. Обеспечение возможности смены статуса заявки с сохранением истории ее исполнения (кр. «Смена статуса заявки»);
   * Описание: В программе должна быть реализована возможность смены статуса заявки. При смене статуса, приложение должно автоматически записывать дату смены статуса заявки;
   * Статус требования: принято;
   * Тестовый пример: От имени работника тех. поддержки Колыпатова Михаила сменить статус заявки Андреева Семёна с выполняется на закрыта.
     + Исходные данные:
       - Колыпатов Михаил авторизован в системе;
       - Текст заявки - “Клавиатура не реагирует на нажатия клавиш”;
       - Текущий статус заявки – «Выполняется»;
       - Время смены статуса заявки - 01.02.2017 12:00:00;
     + Результаты:
       - В приложении изменена заявка, текущие параметры: Дата поступления - 01.02.2017, Дата выполнения - 01.02.2017 12:00:00, Последнее изменение - 01.02.2017 12:00:00, Выполнить до - 02.02.2017, Выполняет - Смирнов Евгений, Статус заявки - закрыта, Текст заявки - Клавиатура не реагирует на нажатия клавиш.
       - Соответствующая запись в БД изменена.
4. Обеспечение возможности смены статуса работы с сохранением истории ее исполнения (кр. «Смена статуса работы»);
   * Описание: В программе должна быть реализована возможность смены статуса работы. При смене статуса, приложение должно автоматически записывать дату смены статуса работы;
   * Статус требования: принято;
   * Тестовый пример: Сменить статус работы назначенной Смирнову Евгению с выполняется на выполнено.
     + Исходные данные:
       - Смирнов Евгений авторизован в системе;
       - Текст заявки - “Клавиатура не реагирует на нажатия клавиш”;
       - Время смены статуса работы - 01.02.2017 11:40:00;
     + Результаты:
       - В приложении изменена работа, текущие параметры: Дата поступления - 01.02.2017, Дата выполнения - 01.02.2017 11:40:00, Последнее изменение - 01.02.2017 11:40:00, Выполнить до - 02.02.2017, Выполняет - Смиирнов Евгений, Статус работы - выполнена, Тип работы - работа с ПК, Текст заявки - Клавиатура не реагирует на нажатия клавиш.
       - Соответствующая запись в БД изменена.
5. Обеспечение возможности разграничения прав пользователей и формирования их списка умений (кр. «Разграничение прав пользователей и формирование умений»);
   * Описание: В программе должна быть реализована возможность разграничения прав пользователей. Пользователь может лишь: создавать заявки и просматривать их статус. Технический исполнитель может лишь: менять статус работы, комментировать работу, просматривать назначенные ему работы и связанные с ними заявки. Специалист тех. поддержки в правах не ограничен. Специалист тех поддержки должен иметь возможность конфигурировать список умений исполнителей под конкретное предприятие;
   * Статус требования: принято;
   * Тестовый пример №1: Пользователь Андреев Семен авторизуется в системе.
     + Исходные данные:
       - Андреев Семен не авторизован в системе;
     + Результаты:
       - Авторизованому пользователю доступны вкладки: «Заявки», «Добавить заявку».
   * Тестовый пример №2: Специалист тех. поддержки Колыпатов Михаил авторизуется в системе и добавляет новый тип работы/умение «Настройка интернета»/ «Выполнение монтажных работ».
     + Исходные данные:
       - Колыпатов Михаил не авторизован в системе;
     + Результаты:
       - Авторизованому пользователю доступны вкладки: «Заявки», «Добавить заявку», «Работы», «Добавить работу», «Конфигурирование списка умений», «Конфигурирование типов работ».
6. Обеспечение формирования типов работ (кр. «Формирование типов работ»).
   * Описание: В программе должна быть реализована возможность форматирования списка типов работ. Специалист тех. поддержки должен иметь возможность конфигурировать список типов работ под конкретное предприятие;
   * Статус требования: принято;
   * Тестовый пример: см. требование 6, Тестовый пример №2.

#### Нефункциональные требования к системе.

* Производительность системы – обеспечить достаточно быстрый поиск и вывод нужной информации для удобства использования;
  + Тестовый пример: Процесс назначение исполнителя на работу не должно занимать больше 7 секунд.
* Безопасность данных – разделения прав доступа к информации в БД;
  + Тестовый пример: Пользователь не должен иметь доступа к заявкам других пользователей;
* Удобство использования – дружественный, интуитивно понятный пользовательский интерфейс;
  + Тестовый пример: Для того чтобы разобраться с интерфейсом системы, сотрудник тех. поддержки должен потратить не более 10 минут.
* Стоимость разработки не должна превышать 15 000 руб;

#### Проектные требования к системе.

* Ограничение ОС Windows 7;
* Пользователь должен иметь навыки работы с компьютером;
* Администратор должен иметь необходимую квалификацию;
* Сервер SQL server;
* Средства разработки на C#.

#### Требования к процессу хранения данных.

Данные приложения, а именно информацию о пользователях, сотрудниках тех. поддержки и исполнителях, заявки пользователей на обслуживание, работы по данным заявкам, а также константные значения, такие как типы работ, статусы заявок/работ и т.п. требуется хранить на удаленном сервере. Хранение данных на сервере должно организовываться с использованием СУБД MS SQL Server.

Предполагается, что приложение будет иметь клиент-серверную архитектуру (рисунок 11).



**Рисунок 11 Клиент-серверная архитектура**

Сотрудник предприятия через приложение на рабочем месте формирует заявку на обслуживание и сохраняет ее, сформированная заявка сохраняется сервером приложение на сервере данных. Отравленная заявка обрабатывается работником отдела тех. обеспечения, формируется список работ соответствующий заявке. По мере выполнения работ добавляются новые записи в БД об изменении статуса работ и заявок.

#### Определение приоритетов требований.

|  |  |
| --- | --- |
| Требование | Приоритет |
| Функциональные | |
| Создание заявки | Реализуется |
| Формирование списка работ | Реализуется |
| Назначение исполнителя на работу | Реализуется |
| Смена статуса заявки | Реализуется |
| Смена статуса работы | Реализуется |
| Разграничение прав пользователей и формирование умений | В проекте |
| Формирование типов работ | В проекте |
| Нефункциональные | |
| Производительность системы | Ограничение |
| Безопасность данных | Ограничение |
| Удобство использования | Ограничение |
| Стоимость разработки | Ограничение |

# Анализ и проектирование

## Определение классов анализа

Выявлены классы:

Классы граничных объектов:

* Класс "Форма" – определён на основе объекта «Форма» сценариев «Создание заявки», «Смена статуса работы», «Работа с заявкой».
* Класс "Форма создания заявки" – определён на основе объекта «Форма» сценария «Создание заявки», является подокном «Формы».
* Класс "Форма работы с заявкой" – определён на основе объекта «Форма» сценария «Работа с заявкой», является подокном «Формы».
* Класс "Форма создания работы" – определён на основе объекта «Форма» сценария «Работа с заявкой», является подокном «Формы работы с заявкой».
* Класс "Форма смены статуса работы" – определён на основе объекта «Форма» сценария «Смена статуса работы», является подокном «Формы».
* Класс "Форма авторизации" - определен на основе объекта «Форма» сценария «Авторизация пользователя», является подокном «Формы».

Классы сущностей:

* Класс "Заявка" – определён на основе объекта «Заявка» сценариев «Работа с заявкой», «Создание заявки». Атрибуты:
  + Идентификатор, тип int - искусственный атрибут, хранит в себе идентификатор объекта;
  + Дата закрытия, тип date - содержит в себе дату закрытия заявки;
  + Дата составления, тип date - содержит в себе дату создания заявки;
  + Дата изменения, тип date - содержит в себе дату последнего изменения заявки;
  + Крайний срок выполнения, тип date - содержит дату до которой заявка должна быть выполнена;
  + Текст заявки, тип char - содержит в себе описание проблемы, возникшей у заявителя;
  + Статус, тип char - содержит в себе текущий статус заявки;
  + Информация о заявителе, тип char - содержит в себе описание заявителя (ФИО);
  + Обработчик заявки, тип char - содержит в себе ФИО специалиста обработавшего заявку.
* Класс "Работа" – определён на основе объекта «Работа» сценария «Смена статуса работы», объекта «Список работ», сценария «Работа с заявкой». Атрибуты:
  + Идентификатор работы, тип int - искусственный атрибут, хранит в себе идентификатор объекта;
  + Дата назначения, тип date - содержит в себе дату создания работы;
  + Дата выполнения, тип date - содержит в себе дату выполнения работы;
  + Дата изменения, тип date - содержит в себе дату последнего изменения работы;
  + Крайний срок выполнения, тип date - содержит дату до которой работа должна быть выполнена;
  + Идентификатор заявки, тип int - искусственный атрибут, хранит в себе идентификатор заявки к которой работа относится;
  + Тип работы, тип char - содержит в себе тип работы;
  + Исполнитель, тип char - содержит в себе описание исполнителя(ФИО);
  + Статус работы, тип char - содержит в себе текущий статус работы;
  + Подтверждение выполнения, тип boolean - показывает, подтверждено ли выполнение работы.
* Класс «Данные пользователей» - определен на основе объекта «Данные пользователей» сценария «Авторизация пользователя». Атрибуты:
  + Идентификатор, тип int - искусственный атрибут, хранит в себе идентификатор пользователя;
  + Логин, тип char - содержит в себе логин пользователя;
  + Пароль, тип char - содержит пароль пользователя.
* Класс «Статусы работ» - определен на основе объекта «Статусы работ» сценария «Смена статуса работы». Атрибуты:
  + Идентификатор, тип int - искусственный атрибут, хранит в себе идентификатор статуса;
  + Наименование, тип char - содержит в себе наименование статуса.
* Класс «Журнал» - определен на основе объекта «Журнал» сценария «Журналирование действий пользователя в системе». Атрибуты:
  + Идентификатор записи, тип int - искусственный атрибут, хранит в себе идентификатор записи;
  + Идентификатор работы, тип int - содержит идентификатор работы в которой сделаны изменения;
  + Инициатор события, тип char - содержит ФИО пользователя, спровоцировавшего создание записи;
  + Дата создания, тип date - содержит дату создания записи;
  + Тип события, тип char - содержит тип события.

Определение методов объектов

Диаграммы последовательности сценариев: «Создание заявки», «Работа с заявкой», «Смена статуса работы».

Диаграмма последовательности сценария «Создание заявки» приведена на рисунке 2.1.

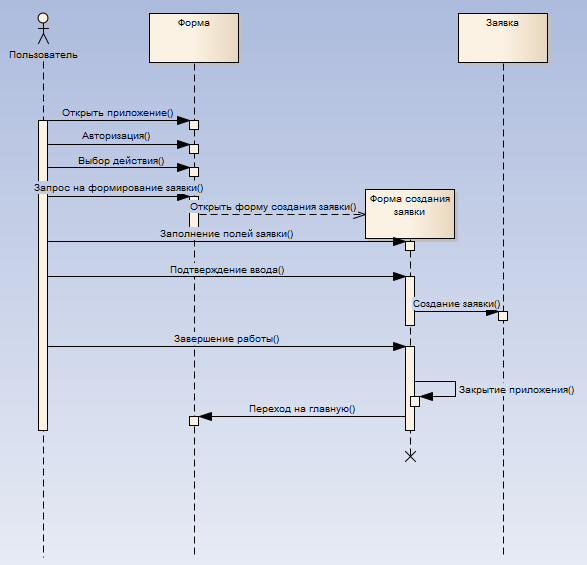


Рисунок 2.1. Создание заявки

В ходе построения объекты диаграммы выявлены следующие объекты:

1. Пользователь
2. Форма
3. Форма создания заявки
4. Заявка

Выявлены следующие сообщения:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| № | Вызывающий объект | Сообщение | Получатель сообщения | Примечание |
| 1 | Пользователь | Открыть приложение | Форма |  |
| 2 | Пользователь | Авторизация | Форма |  |
| 3 | Пользователь | Выбор действия | Форма |  |
| 4 | Пользователь | Запрос на формирование заявки | Форма |  |
| 5 | Форма | Открыть форму создания заявки | Форма создания заявки |  |
| 6 | Пользователь | Заполнить поля заявки | Форма создания заявки |  |
| 7 | Пользователь | Подтверждение ввода | Форма создания заявки |  |
| 8 | Форма создания заявки | Создать заявку | Заявка |  |
| 9 | Пользователь | Завершить работу | Форма создания заявки |  |
| 10 | Форма создания заявки | Закрыть форму | Форма создания заявки |  |
| 11 | Форма создания заявки | Перейти на главную | Форма |  |

|  |  |
| --- | --- |
| Операторы | Список сообщений |
| Форма | Открыть форму создания заявки. |
| Форма создания заявки | Создать заявку;  Закрыть форму;  Вернуться на главную. |

Диаграмма последовательности сценария «Работа с заявкой» приведена на рисунках 2.2 и 2.3.

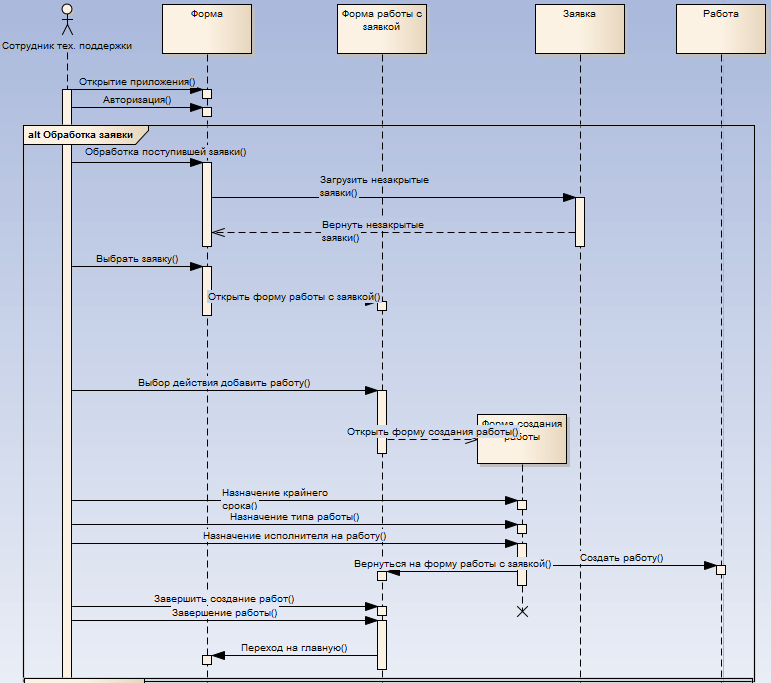


Рисунок 2.2 Работа с заявкой (Обработка заявки)



Рисунок 2.3. Работа с заявкой (Закрытие заявки)

В ходе построения объекты диаграммы выявлены следующие объекты:

1. Сотрудник тех. поддержки
2. Форма
3. Форма работы с заявкой
4. Заявка
5. Форма создания работы
6. Работа

Выявлены следующие сообщения:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| № | Вызывающий объект | Сообщение | Получатель сообщения | Примечание |
| 1 | Сотрудник тех. поддержки | Открытие приложения | Форма |  |
| 2 | Сотрудник тех. поддержки | Авторизация | Форма |  |
| Сценарий добавления работ к заявке | | | | |
| 3.1 | Сотрудник тех. поддержки | Обработать поступившую заявку | Форма |  |
| 4.1 | Форма | Загрузить незакрытые заявки | Заявка |  |
| 5.1 | Заявка | Вернуть незакрытые заявки | Форма |  |
| 6.1 | Сотрудник тех. поддержки | Выбрать заявку | Форма |  |
| 7.1 | Форма | Открыть форму работы с заявкой | Форма работы с заявкой |  |
| 8.1 | Сотрудник тех. поддержки | Выбрать действие добавить работу | Форма работы с заявкой |  |
| 9.1 | Форма работы с заявкой | Открыть форму создания работы | Форма создания работы |  |
| 10.1 | Сотрудник тех. поддержки | Назначить крайний срок | Форма создания работы |  |
| 11.1 | Сотрудник тех. поддержки | Назначить тип работы | Форма создания работы |  |
| 12.1 | Сотрудник тех. поддержки | Назначить исполнителя на работу | Форма создания работы |  |
| 13.1 | Форма создания работы | Создать работу | Работа |  |
| 14.1 | Форма создания работы | Вернуться на форму работы с заявкой | Форма работы с заявкой |  |
| 15.1 | Сотрудник тех. поддержки | Завершить создание работ | Форма работы с заявкой |  |
| 16.1 | Сотрудник тех. поддержки | Завершить работу | Форма работы с заявкой |  |
| 17.1 | Форма работы с заявкой | Перейти на главную | Форма |  |
| Сценарий закрытия заявки | | | | |
| 3.2 | Сотрудник тех. поддержки | Обработать текущую заявку | Форма |  |
| 4.2 | Форма | Загрузить незакрытые заявки | Заявка |  |
| 5.2 | Заявка | Вернуть незакрытые заявки | Форма |  |
| 6.2 | Сотрудник тех. поддержки | Выбрать заявку | Форма |  |
| 7.2 | Форма | Открыть форму работы с заявкой | Форма работы с заявкой |  |
| 8.2 | Сотрудник тех. поддержки | Закрыть заявку | Форма работы с заявкой |  |
| 9.2 | Форма работы с заявкой | Обновить статус заявки | Заявка |  |
| 10.2 | Сотрудник тех. поддержки | Завершить работу | Форма работы с заявкой |  |
| 11.2 | Форма работы с заявкой | Перейти на главную | Форма |  |

|  |  |
| --- | --- |
| Операторы | Список сообщений |
| Форма | Загрузить закрытые заявки;  Открыть форму работы с заявкой. |
| Форма работы с заявкой | Открыть форму создания работы;  Обновить статус заявки;  Перейти на главную. |
| Форма создания работы | Создать работу;  Вернуться на форму работы с заявкой. |

Диаграмма последовательности сценария «Смена статуса работы» приведена на рисунке 2.4.

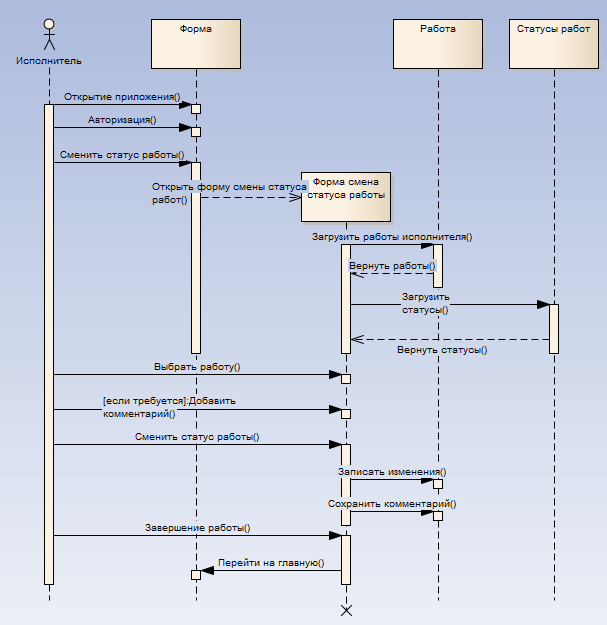


Рисунок 2.4. Смена статуса работы

В ходе построения объекты диаграммы выявлены следующие объекты:

1. Исполнитель
2. Форма
3. Форма смены статуса работы
4. Работа
5. Статусы работ

Выявлены следующие сообщения:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| № | Вызывающий объект | Сообщение | Получатель сообщения | Примечание |
| 1 | Исполнитель | Открытие приложения | Форма |  |
| 2 | Исполнитель | Авторизация | Форма |  |
| 3 | Исполнитель | Сменить статус работы | Форма |  |
| 4 | Форма | Открыть форму смены статуса работы | Форма смены статуса работы |  |
| 5 | Форма смены статуса работы | Загрузить работы исполнителя | Работа |  |
| 6 | Работа | Вернуть работы | Форма смены статуса работы |  |
| 7 | Форма смены статуса работы | Загрузить статусы работ | Статусы работ |  |
| 8 | Статусы работ | Вернуть статусы | Форма смены статуса работы |  |
| 9 | Исполнитель | Выбрать работу | Форма смены статуса работы |  |
| 10 | Исполнитель | Добавить комментарий | Форма смены статуса работы | Если требуется |
| 11 | Исполнитель | Сменить статус работы | Форма смены статуса работы |  |
| 12 | Форма смены статуса работы | Записать изменения | Работа |  |
| 13 | Форма смены статуса работы | Сохранить комментарий | Работа |  |
| 14 | Исполнитель | Завершить работу | Форма смены статуса работы |  |
| 15 | Форма смены статуса работы | Перейти на главную | Форма |  |

|  |  |
| --- | --- |
| Операторы | Список сообщений |
| Форма | Открыть форму смены статуса работы. |
| Форма смены статуса работы | Загрузить работы исполнителя;  Загрузить статусы;  Записать изменения;  Сохранить комментарий;  Перейти на главную. |

Диаграмма последовательности сценария «Авторизация» приведена на рисунке 2.5.

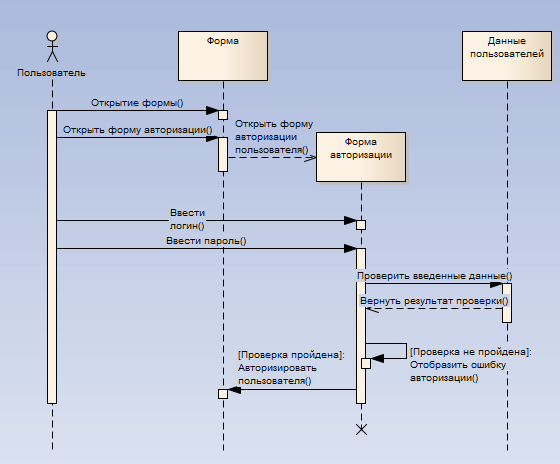


Рисунок 2.5. Авторизация

В ходе построения объекты диаграммы выявлены следующие объекты:

1. Исполнитель
2. Форма
3. Форма авторизации
4. Данные пользователя

Выявлены следующие сообщения:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| № | Вызывающий объект | Сообщение | Получатель сообщения | Примечание |
| 1 | Исполнитель | Открытие приложения | Форма |  |
| 2 | Исполнитель | Открыть форму авторизации | Форма |  |
| 3 | Форма | Открыть форму авторизации пользователя | Форма авторизации |  |
| 4 | Исполнитель |  | Форма авторизации |  |
| 5 | Исполнитель | Загрузить работы исполнителя | Форма авторизации |  |
| 6 | Форма авторизации | Проверить введенные данные | Данные пользователей |  |
| 7 | Данные пользователей | Вернуть результаты проверки | Форма авторизации |  |
| 8 | Форма авторизации | Отобразить ошибку авторизации | Форма авторизации | Проверка не пройдена |
| 9 | Форма авторизации | Авторизовать пользователя | Форма | Проверка пройдена |

|  |  |
| --- | --- |
| Операторы | Список сообщений |
| Форма | Открыть форму авторизации. |
| Форма авторизации | Проверить введенные данные;  Отобразить ошибку авторизации;  Авторизовать пользователя. |

Диаграмма последовательности сценария «Журналирование пользователей в системе» приведена на рисунке 2.6.

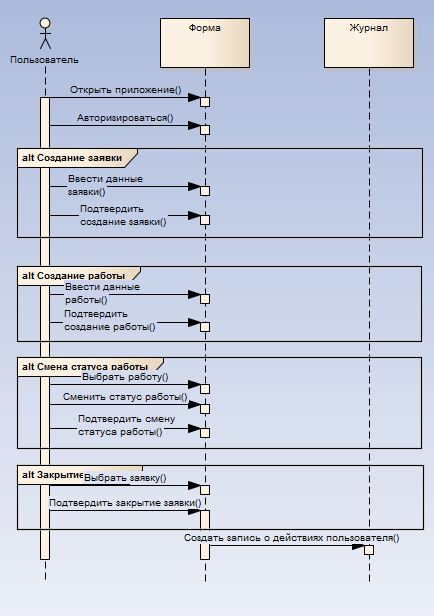


Рисунок 2.6. Авторизация

В ходе построения объекты диаграммы выявлены следующие объекты:

1. Пользователь
2. Форма
3. Журнал

Выявлены следующие сообщения:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| № | Вызывающий объект | Сообщение | Получатель сообщения | Примечание |
| 1 | Пользователь | Открытие приложения | Форма |  |
| 2 | Пользователь | Авторизация | Форма |  |
| Создание заявки | | | | |
| 3.1 | Пользователь | Ввести данные заявки | Форма |  |
| 3.2 | Пользователь | Подтвердить создание заявки | Форма |  |
| Создание работы | | | | |
| 4.1 | Пользователь | Ввести данные работы | Форма |  |
| 4.2 | Пользователь | Подтвердить создание работы | Форма |  |
| Смена статуса работы | | | | |
| 5.1 | Пользователь | Выбрать работу | Форма |  |
| 5.2 | Пользователь | Сменить статус работы | Форма |  |
| 5.3 | Пользователь | Подтвердить смену статуса работы | Форма |  |
| Закрытие заявки | | | | |
| 6.1 | Пользователь | Выбрать заявку | Форма |  |
| 6.2 | Пользователь | Подтвердить закрытие заявки | Форма |  |
| 7 | Форма | Создать запись о действиях пользователя | Журнал |  |

|  |  |
| --- | --- |
| Операторы | Список сообщений |
| Форма | Создать запись о действиях пользователя. |

Выявленные методы классов анализа:

* Класс "Форма"
  + Открыть форму смены статуса работ, тип возврата void - создает граничный объект класса «Форма смены статуса работ»;
  + Открыть форму создания заявки, тип возврата void - создает граничный объект класса «Форма создания заявки»;
  + Открыть форму работы с заявкой, тип возврата void - создает граничный объект класса «Форма работы с заявкой»
  + Загрузить незакрытые заявки, тип возврата List <Заявка> - возвращает список оставленных пользователями заявок.
* Класс "Форма создания заявки"
  + Создать заявку, тип возврата Заявка – создает заявку на основе введенных данных;
  + Закрыть форму, тип возврата void – завершает работу на форме;
  + Перейти на главную, тип возврата void - осуществляет переход на главную страницу приложения.
* Класс "Форма работы с заявкой"
  + Открыть форму создания работы, тип возврата void – создает граничный объект класса «Форма создания работы»;
  + Перейти на главную, тип возврата void - осуществляет переход на главную страницу приложения.
  + Обновить статус заявки, тип возврата void - записывает изменения произошедшие с заявкой в БД.
* Класс "Форма создания работы"
  + Создать работу, тип возврата Работа – создает работу на основе введенных данных.
  + Вернуться на форму работы с заявкой, тип возврата void - осуществляет переход на страницу работы с заявкой.
* Класс "Форма смены статуса работы"
  + Записать изменения, тип возврата void – сохраняет изменения сделанные в работе;
  + Загрузить работы исполнителя, тип возврата List <Работа>– загружает из БД работы назначенные исполнителю;
  + Сохранить комментарий, тип возврата void - сохраняет оставленный исполнителем комментарий к работе;
  + Перейти на главную, тип возврата void - осуществляет переход на главную страницу приложения.
* Класс "Форма авторизации"
  + Проверить введенные данные, тип возврата void - запрашивает проверку логина и пароля пользователя;
  + Авторизовать пользователя, тип возврата void - авторизует пользователя в системе;
  + Отобразить ошибку авторизации, тип возврата void - отображает ошибку авторизации.

Диаграмма классов-анализа приведена на рисунке 2.7:

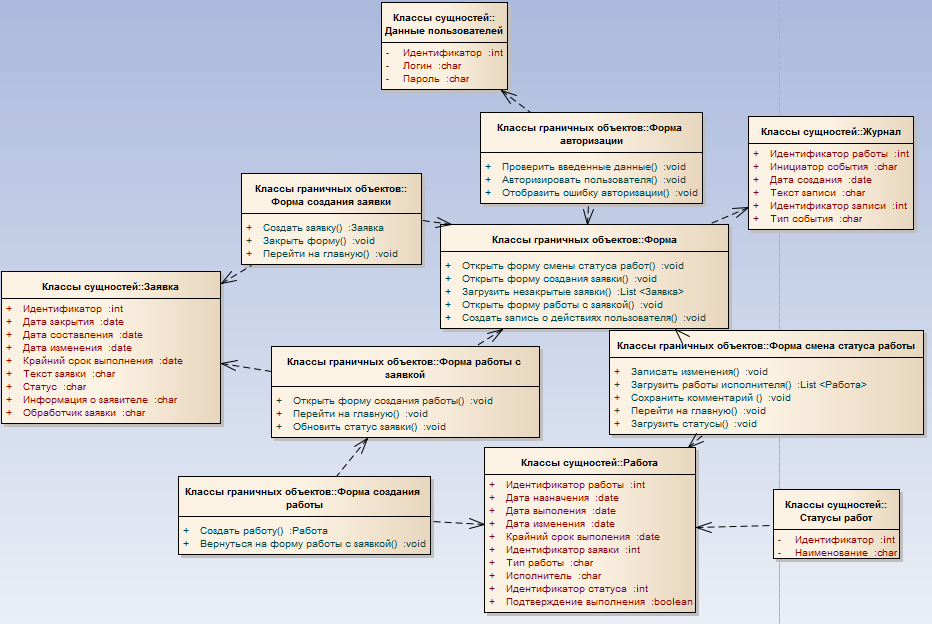


Рисунок 2.7. Диаграмма классов-анализа.

# Разработка системы централизованного хранения и обработки данных

## Формирование требований к БД

На основе требований заказчика система должна быть построена на основе СУБД поддерживающей архитектуру клиент-сервер. Приложение разрабатывается с использованием технологий ASP.NET MVC и Entity Framework.

База данных должна хранить информацию о пользователях системы, блюдах, ингредиентах, меню, поставках, а также их составах, типах блюд и требованиях к меню. База данных должна быть универсальной, т.е. оставаться работоспособной на любой машине и подстраиваться под видоизменение пользовательского интерфейса, быть способной к реорганизации и расширения при изменении предметной области.

В качестве СУБД для разработки системы выбрана Microsoft SQL Server 2012. Microsoft SQL Server — система управления реляционными базами данных (СУБД), разработанная корпорацией Microsoft. Используется для работы с базами данных размером от персональных до крупных баз данных масштаба предприятия. Широкий спектр использования Microsoft SQL Server обосновывает целесообразность выбора.

## Формирование отношений БД

На основе классов, выявленных в разделе анализа и проектирования ИС, были определены следующие сущности базы данных:



На основе классов, выявленных в разделе анализа и проектирования ИС, были определены следующие сущности базы данных:

* + Сущность «Entrance» - определена на основе класса проектирования «Поставка», отражает поставки ингредиентов на склад;
  + Сущность «EntranceComposition» - связующая таблица, используется для организации связи «многие ко многим» между сущностями «Entrance» и «Ingredient».
  + Сущность «Ingredient» - справочная сущность, определена на основе класса проектирования «Ингредиент», ингредиенты, используемые в блюдах;
  + Сущность «DishComposition» - связующая таблица, используется для организации связи «многие ко многим» между сущностями «Dish» и «Ingredient»;
  + Сущность «Dish» - справочная сущность, определена на основе класса проектирования «Блюдо», хранит в себе все возможные блюда;
  + Сущность «MenuComposition» - связующая таблица, используется для организации связи «многие ко многим» между сущностями «Dish» и «Menu»;
  + Сущность «Menu» - определена на основе класса проектирования «Меню», хранит информацию о меню;
  + Сущность «MenuRequirement» - определена на основе класса проектирования «Требования к меню», хранит информацию о требованиях к меню;
  + Сущность «DishTypes» - справочная сущность, хранит в себе все возможные типы блюд.
  + Сущность «AspNetUsers» - определена на основе класса проектирования «Данные пользователей», хранит в себе данные пользователей, применяемые при авторизации;
  + Сущность «AspNetRoles» - справочная сущность, хранит в себе все возможные уровни доступа пользователей к системе.
  + Сущность «AspNetUserRoles» - связующая таблица, используется для организации связи «многие ко многим» между сущностями «AspNetUsers» и «AspNetRoles».

## Даталогическое проектирование

В связи с тем, что для разработки будет использована технология MVC, таблицы базы данных будут сформированы автоматически, на основе определенных классов анализа.

На основе состава сущностей, созданы таблицы БД:

* " Entrance " – хранит в себе поставки. Атрибуты:
  + Id, тип int – искусственный атрибут, хранит в себе идентификатор поставки;
  + Date, тип datetime – содержит дату поставки;
  + Recalculation, тип bit – показывает пересчитывались ли цены ингредиентов из этой поставки.

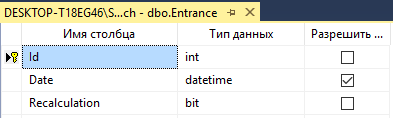


Рисунок 3.1. Таблица Entrance

* " EntranceComposition" – хранит в себе состав поставки. Атрибуты:
  + Id, тип int – искусственный атрибут, хранит в себе идентификатор состава поставки;
  + Entrance, тип int – хранит в себе идентификатор поставки, к которой относится состав;
  + Ingredient, тип int - хранит в себе идентификатор ингредиента, включенного в поставку;
  + Count, тип float – хранит в себе количество пришедшего ингредиента;
  + Cost, тип money – хранит в себе стоимость закупленного ингредиента.

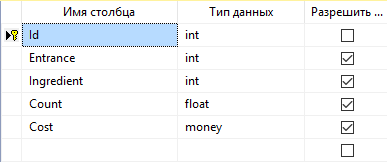


Рисунок 3.2. Таблица EntranceComposition

* " Ingredient " – хранит в себе возможные ингредиенты. Атрибуты:
  + Id, тип int – искусственный атрибут, хранит в себе идентификатор ингредиента;
  + Name, тип char - содержит наименование ингредиента;
  + Cost, тип money – хранит в себе стоимость ингредиента;
  + Count, тип float – хранит в себе количество ингредиента на складе.

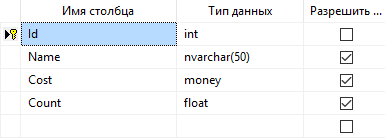


Рисунок 3.3. Таблица Ingredient

* "DishComposition" - хранит в себе состав блюд.
  + Id, тип int – искусственный атрибут, хранит в себе идентификатор состава блюд;
  + Dish, тип int – хранит в себе идентификатор блюда, к которому относится состав;
  + Ingridient, тип int - хранит в себе идентификатор ингредиента, используемого в блюде;
  + Count, тип float - хранит в себе необходимое количество ингредиента;

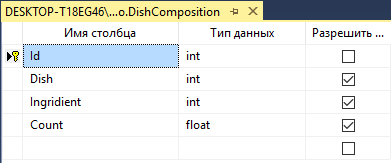


Рисунок 3.4. Таблица DishComposition

* " Dish " - хранит возможные блюда. Атрибуты:
  + Id, тип int - искусственный атрибут, хранит в себе идентификатор блюда;
  + Name, тип char – хранит в себе название блюда;
  + Type, тип int - хранит в себе идентификатор типа блюд, к которому относится блюдо;
  + Cost, тип money – хранит в себе стоимость блюда;
  + Frequency, тип int – хранит в себе частоту попадания блюда в меню;

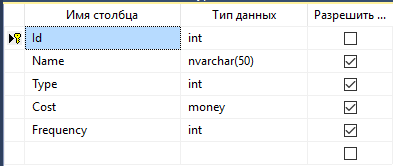


Рисунок 3.5. Таблица Dish

* MenuComposition - хранит в себе состав меню. Атрибуты:
  + Id, тип int - искусственный атрибут, хранит в себе идентификатор состава меню;
  + Menu, тип int – хранит в себе идентификатор меню, к которому относится состав;
  + DishComposition, тип int – хранит в себе идентификатор блюда, включаемого в меню;



Рисунок 3.6. Таблица MenuComposition

* AspNetUsers - автосгенерированная таблица, хранит в себе информацию о персонах, работающих в КОМК. Атрибуты:
  + Id, тип int - искусственный атрибут, хранит в себе идентификатор персоны;
  + UserName, тип nvarchar - содержит логин сотрудника (по факту Email);
  + Email, тип char - содержит Email сотрудника;
  + PasswordHash, тип nvarchar - содержит зашифрованный пароль сотрудника.
  + SecurityStamp, тип nvarchar - содержит “Соль” для пароля сотрудника.

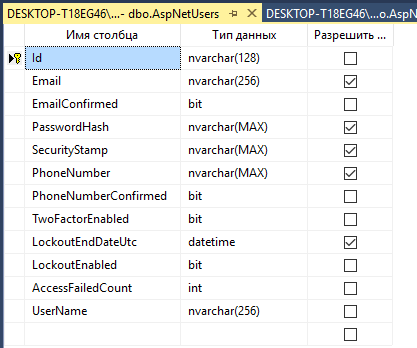


Рисунок 3.7. Таблица AspNetUsers

* AspNetRoles - хранит в себе информацию о ролях сотрудников, работающих в КОМК. Атрибуты:
  + Id, тип nvarchar
  + RoleId, тип nvarchar - искусственный атрибут, хранит в себе идентификатор роли;
  + Description, тип nvarchar - содержит наименование роли.

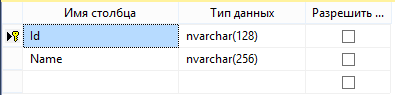


Рисунок 3.8. Таблица AspNetRoles

* AspNetUserRoles- таблица, обеспечивающая связь многие к многим между ролями и пользователями. Атрибуты:
  + RoleId, тип nvarchar- искусственный атрибут, хранит в себе идентификатор роли;
  + UserId, тип nvarchar- искусственный атрибут, хранит в себе идентификатор персоны;
  + UserRoleId, тип int - искусственный атрибут, хранит в себе идентификатор записи.

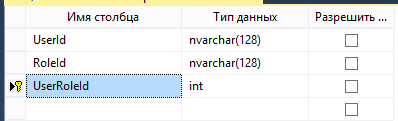


Рисунок 3.9. Таблица AspNetUserRoles

* Menu - хранит в себе меню. Атрибуты:
  + Id, тип int - искусственный атрибут, хранит в себе идентификатор меню;
  + Date, тип date – содержит дату, на которую составлено меню;

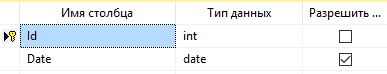


Рисунок 3.10. Таблица Menu

* "MenuRequirement" – хранит в себе требования к составу меню. Атрибуты:
  + Id, тип int - искусственный атрибут, хранит в себе идентификатор требований
  + Menu, тип int - содержит идедификатор меню, к которому относятся требования;
  + DishType, тип int - содержит идентификатор типа блюда, которое должно быть в меню;
  + Count, тип int – содержит количество блюд данного типа.

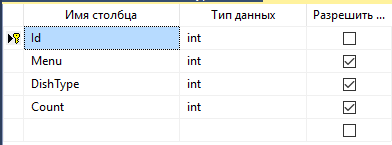


Рисунок 3.11. Таблица MenuRequirement

* "DishType" – хранит в себе типы блюд. Атрибуты:
  + Id, тип int – искусственный атрибут, хранит в себе идентификатор типа блюд;
  + Name, тип char - содержит наименование типа блюд.

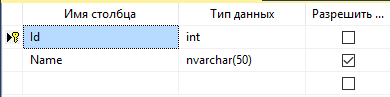


Рисунок 3.12. Таблица DishType

## 3.4 Разработка сценариев работы с данными

* Добавление данных:
  + Добавление блюд. При создании нового блюда в таблицу Dish, добавляется новая запись, автоматически генерируется ее идентификатор Id, в поле Name заносится название блюда, в поле Type заносится идентификатор типа блюда, в поля Frequency и Cost автоматически добавляется значение «0».
  + Добавление ингредиентов. При создании ингредиента в таблицу Ingredient добавляется новая запись, автоматически генерируется ее идентификатор Id, в поле Name заносится название ингредиента, в поле Cost заносится предполагаемая цена ингредиента, в поле Count автоматически добавляется значение «0».
  + Добавление меню. При создании меню в таблицу Menu добавляется новая запись, автоматически генерируется ее идентификатор Id, в поле Date заносится дата, на которую создано меню.
  + Добавление требований к меню. При добавлении требований в таблицу MenuRequirement добавляется новая запись, автоматически генерируется ее идентификатор Id, в поле Menu заносится идентификатор меню, в поле DishType заносится идентификатор типа блюд, в поле Count заносится необходимое количество блюд выбранного типа.
  + Добавление состава меню. При добавлении состава меню в таблице MenuComposition добавляется новая запись, автоматически генерируется ее идентификатор Id, в поле Menu заносится идентификатор меню, в поле DishComposition добавляется блюдо, тип которого указан в требовании к меню с минимальным значением поля Frequency.
  + Добавление поставки. При создании поставки в таблицу Entrance добавляется новая запись, автоматически генерируется ее идентификатор Id, в поле Date заносится дата прихода поставки, в поле Recalculation автоматически добавляется значение «false».
  + Добавление состава поставки. При добавлении состава поставки в таблицу EntranceComposition добавляется новая запись, автоматически генерируется ее идентификатор Id, в поля Entrance и Ingredient добавляются идентификаторы поставки и ингредиента соответственно, в поля Count и Cost добавляются цена ингредиента и его количество в поставке.
  + Добавление типа блюд. При создании типа блюд в таблицу DishTypes добавляется новая запись, автоматически генерируется ее идентификатор Id, в поле Name заносится название типа.
  + Добавление состава блюд. При создании состава блюд в таблицу DishComposition добавляется новая запись, автоматически генерируется ее идентификатор Id, в поля Ingridient и Dish заносятся идентификаторы ингредиента и блюда, в поле Count заносится количество необходимого ингредиента.
  + Добавление пользователей. При создании нового пользователя в таблицу AspNetUsers добавляется новая запись, автоматически генерируется ее идентификатор Id, в поле Email и UserName записывается email пользователя (они будут использоваться для авторизации), в поле PasswordHash записывается пароль введенный пользователем в хешированном виде, в поле SecurityStamp записывается автоматически сгенерированная “соль” к паролю.
* Изменение данных:
  + Пересчет стоимости блюда. В выбранной записи из таблицы Dish в поле Cost меняется значение на сумму стоимостей связанных ингредиентов, умноженных на количество, необходимое для приготовления блюда.
  + Пересчет стоимости ингредиентов. Выбирается запись из таблицы Entrance, находятся связанные с ней записи в таблице EntranceComposition, затем находятся связанные с последней записи в таблице Ingredient, в двух последних таблицах поля Cost и Count перемножаются и суммируются, затем делятся на сумму полей Count. Полученное значение заносится в поле Cost таблицы Ingredient, а в поле Count инкрементируется полем Count таблицы EntranceComposition.
* Удаление данных:
  + Удаление состава поставки. Происходит поиск составляющей поставки из таблицы по идентификатору и удаление всей записи.
  + Удаление состава меню и требований к меню происходит аналогичным образом.

## 3.5 Реализация сценариев работы с данными

* Работа с данными на уровне БД:

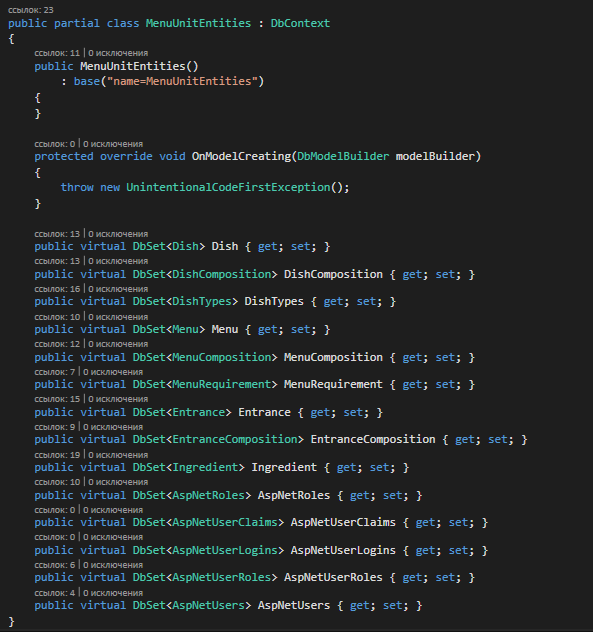
Работы с базой данных происходит только на стороне клиента.

* Работа с данными на уровне клиента:

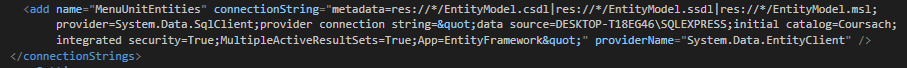
В любом приложении, работающим с БД через Entity Framework, нам нужен будет контекст (класс производный от DbContext) и набор данных DbSet, через который мы сможем взаимодействовать с таблицами из БД. В данном случае таким контекстом является класс MenuUnitEntities.

DbContext: определяет контекст данных, используемый для взаимодействия с базой данных.

DbSet<TEntity>: представляет набор сущностей, хранящихся в базе данных.



И теперь нам надо установить подключение к базе данных. Для установки подключения обычно используется файл конфигурации приложения. В проектах для десктопных приложений файл конфигурации называется *App.config* (как в нашем случае), в проектах веб-приложений - *web.config*. В нашем случае это файл *Web.config*. После добавления Entity Framework он выглядит примерно следующим образом:



Теперь чтобы работать с данными в любой части программы просто создаем элемент контекста и работаем с ним.



* **Сценарии добавления данных**

Рассмотрим добавление новой записи в таблицу БД (например, в таблицу Entrance).

Получаем из представления значения полей сущности entrance, проверяем валидность введенных пользователем данных, добавляем сущность в таблицу Entrance и сохраняем изменения в БД.

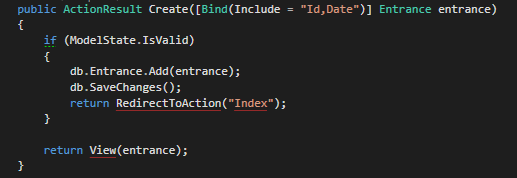


Рисунок 3.13 Добавление данных в таблицу Entrance

Алгоритмы добавления блюда, ингредиента, меню, типа блюд, поставки, состава поставки и требований к меню работают аналогичным образом.

* **Сценарии изменения данных**

Рассмотрим алгоритм пересчета стоимости блюда. Производится поиск блюда в контексте по идентификатору изменяемого блюда и загружаются все его компоненты. Новая стоимость блюда складывается из стоимости компонент умноженной на их количество. Состояние блюда помечается измененным, затем изменения сохраняются в БД.

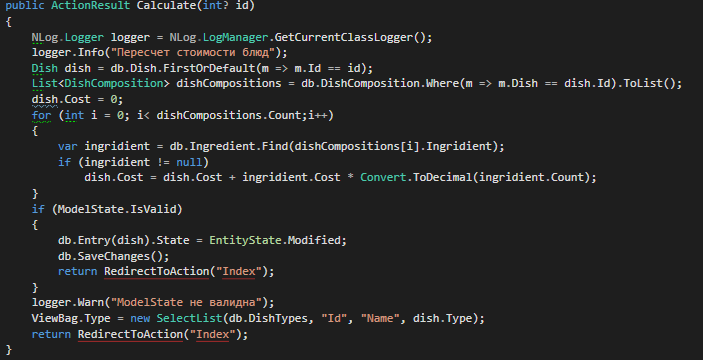


Рисунок 3.14 Изменение данных в таблице Dish

Алгоритм изменения ингредиента работает аналогичным образом.

* **Сценарии удаления данных**

Все сценарии удаления реализованы идентично:

С помощью объекта контекста производим поиск записи из таблицы по параметру id. Найденную запись удаляем из таблицы и сохраняем изменения в базе данных.

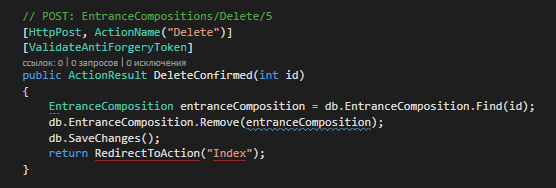


Рисунок 3.15 Удаление данных из БД

# Специальная часть.

Тестирование - один из важнейших этапов разработки приложения. Задача тестирования, проверить реализованы ли требования к разрабатываемому приложению, описанные в техническом задании.

Базовый метод тестирования, это непосредственный запуск приложения и проверка его функции самим разработчиком или тестировщиком. Еще одним возможным методом тестирования является создание в приложении специальных классов, отвечающих за тестирование правильности работы бизнес-логики, описанной в приложении. Такие классы-тестеры называю юнит-тестами, а сам процесс тестирования модульным тестированием приложения.

## Модульное тестирование приложений.

* Описание технологии.

Модульное тестирование (Юнит-тестирование, Unit testing) – процесс в программировании, позволяющий проверить на корректность отдельные модули исходного кода программы.

Основная идея модульного тестирования, писать тесты не для приложения в целом, а для отдельных его методов и классов. Это позволяет достаточно быстро проверять, не привело ли изменение кода в методе класса к регрессии, т.е. не вызвало ли изменение кода ошибок в уже рабочих местах программы. Модульное тестирование облегчает обнаружение и исправление подобных ошибок

Преимущества модульного тестирования:

* Поощрение изменений – модульное тестирование позволяет разработчику свободно менять код тестируемых методов и тут же тестировать измененный
* Упрощение интеграции – может быть использовано для подхода к тестированию «снизу вверх», т.е. сначала тестируются отдельные функции приложения, затем происходит плавный переход к тестированию приложения в целом;
* Документирование кода – модульные тесты можно рассматривать как своеобразное описание работы тестируемого класса;
* Пример использования.

Для реализации модельного тестирования в среде разработки Visual Studio нужно добавить в тестируемое решение отдельный проект «Проект модульного теста» (англ. Unit Test Project) (рисунок 3.1) .

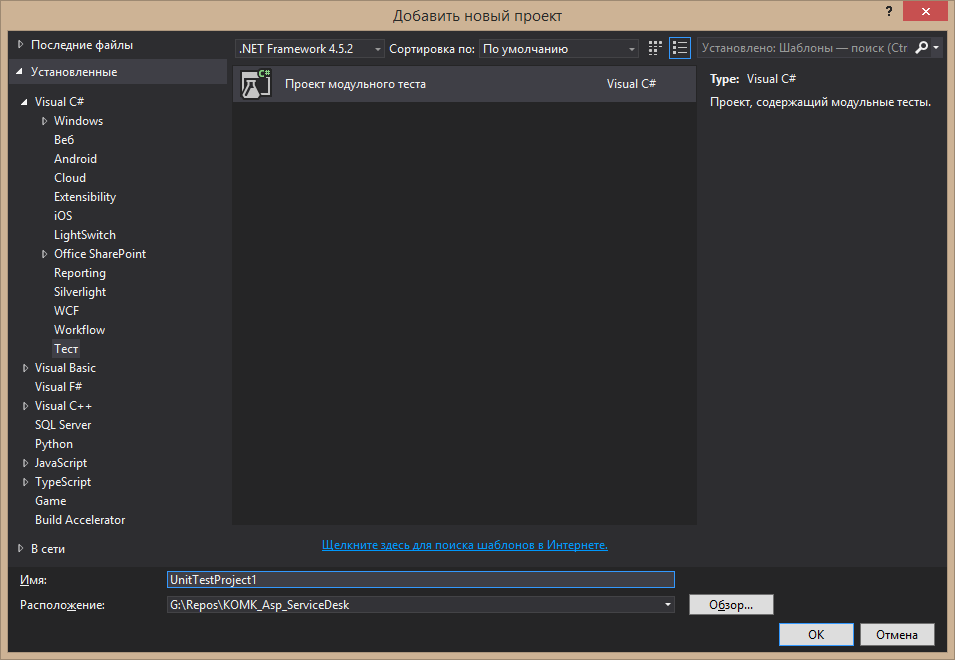


Рисунок 4.1 Добавление в решение проекта модульных тестов.

Альтернативный способ – зайти в код класса, тестирование которого требуется провести, нажать правой кнопкой на имя класса и выбрать пункт «Создание модульных тестов» (рисунок 3.2), Visual Studio автоматически создаст в решении проект модульного теста.

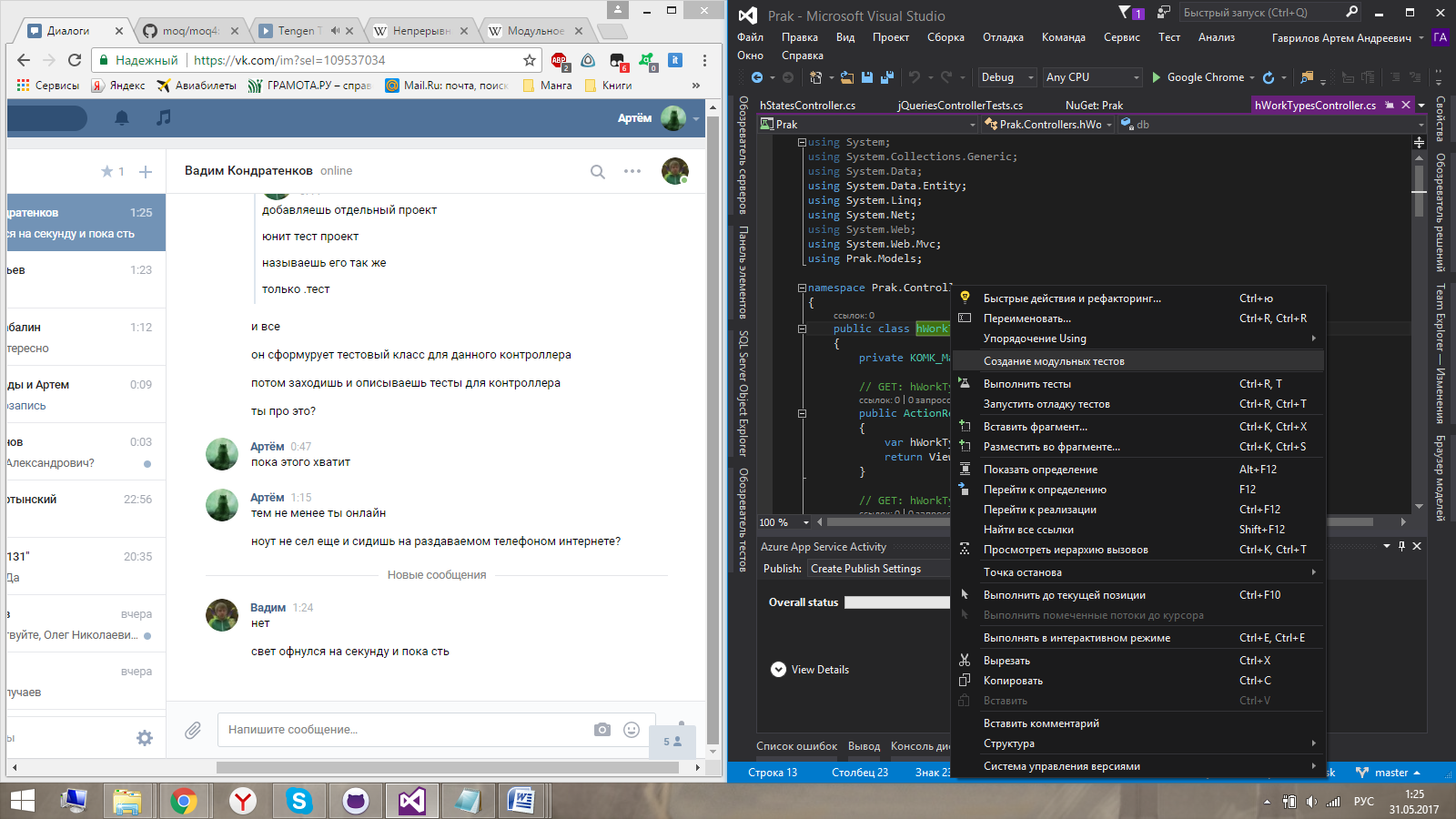


Рисунок 4.2 Добавление в решение проекта модульных тестов.

Воспользуемся вторым методом. Дерево созданного проекта модульных тестов представлено на рисунке 3.3.

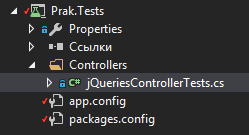


Рисунок 4.3 Дерево проекта модульных тестов.

Протестируем метод контроллера jQueriesController GetAdmin(), возвращающий список пользователей с ролью администратора (он же сотрудник отдела тех. поддержки). Код тестов представлен на рисунках 3.4, 3.5. Код метода GetAdmin() представлен на рисунке 3.6.

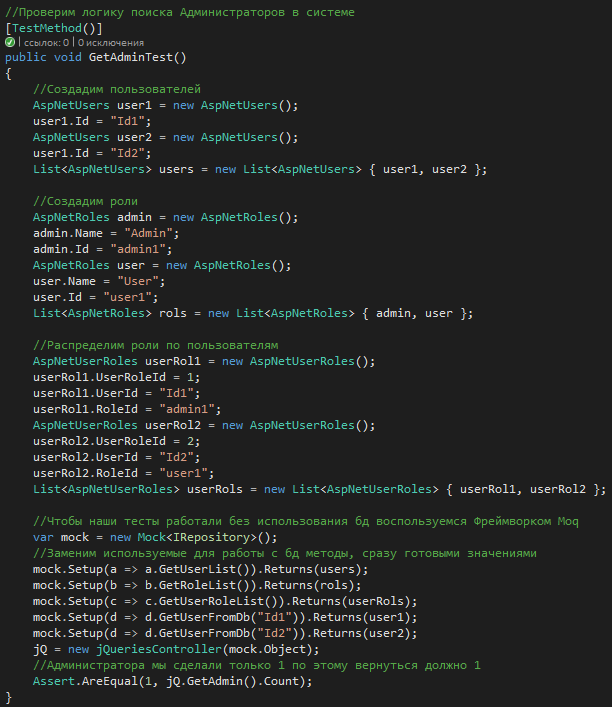


Рисунок 4.4 Оптимистичное тестирование метода GetAdmin()

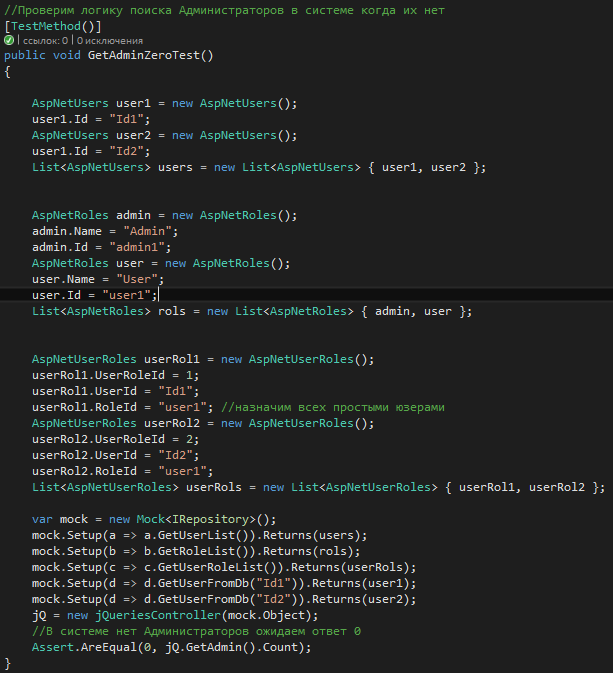


Рисунок 4.5 Пессимистичное тестирование метода GetAdmin()

Для того что бы полностью протестировать метод было написано два теста: оптимистичный, который проверяет ситуацию, когда в БД есть администратор, и пессимистичный, для ситуации когда метод не должен обнаружить администратора в системе.

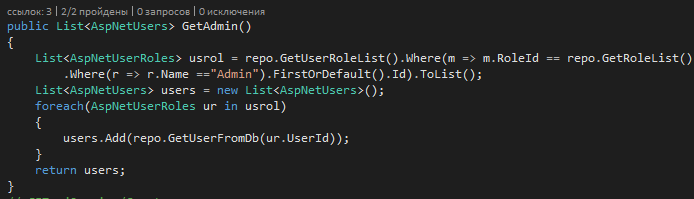


Рисунок 4.6 Код тестируемого метода

Для проведения тестов перейти к соответсвующей команде «Тест -> Выполнить -> Все тесты» (рисунок 3.7).

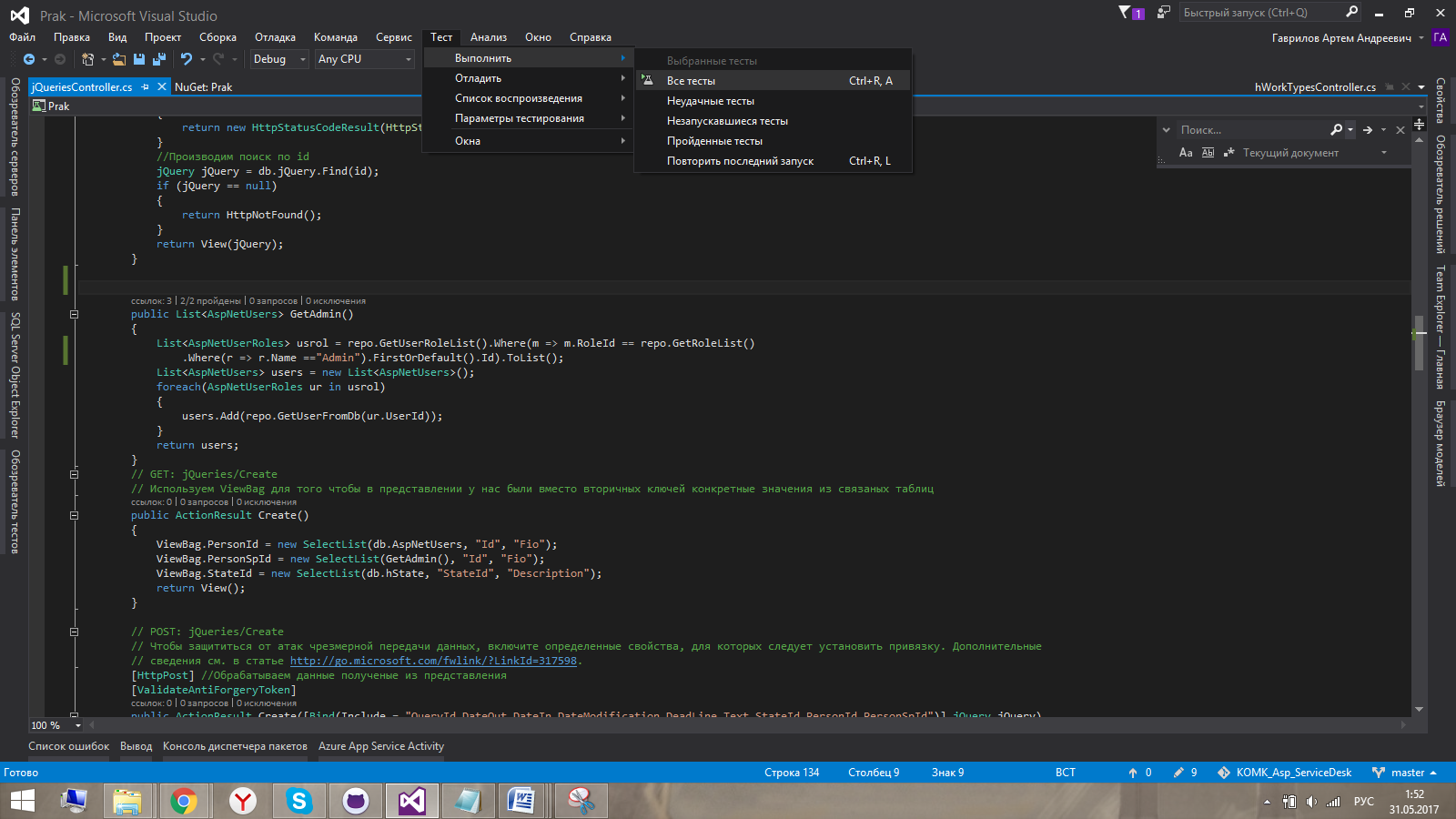


Рисунок 4.7 Запуск тестов.

Результат выполнения тестов можно посмотреть в «Обозревателе тестов» (рисунок 3.8).

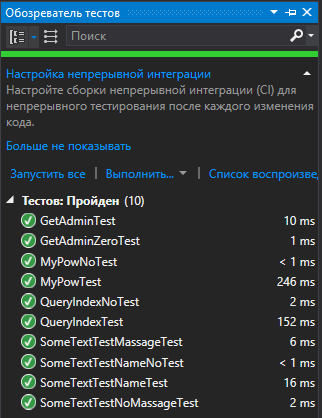


Рисунок 4.8 Результат проведения тестов.

## Автоматизация процесса тестирования с использованием веб-сервиса AppVeyor.

* **Описание технологии.**

Непрерывная интеграция (англ. Continuous Integration) – практика разработки ПО, заключающаяся в слиянии рабочих копий в общую основную ветвь разработки несколько раз в день и выполнении частых автоматизированных сборок проекта для скорейшего выявления и решения интеграционных проблем.

Преимущества:

* Проблемы интеграции выявляются и исправляются быстро, что оказывается дешевле;
* Немедленный прогон модульных тестов для свежих изменений;
* Постоянное наличие текущей стабильной версии вместе с продуктами сборок — для тестирования, демонстрации, и т. п.

Недостатки:

* Затраты на поддержку работы непрерывной интеграции;
* Потенциальная необходимость в выделенном сервере под нужды непрерывной интеграции.

Требования к проекту:

* Проект должен храниться в репозитории системы управления версиями;
* Операции копирования из репозитория, сборки и тестирования всего проекта автоматизированы и легко вызываются из внешней программы.

Существует специальные сервисы, реализующие данную практику. Среди них:

* Travis CI;
* AppVeyor;
* GitLab CI.
* **Пример использования.**

Воспользуемся сервисом AppVeyor.

Перейдем по ссылке <https://ci.appveyor.com/login> и зайдем в него воспользовавшись своим аккаунтом на githab.

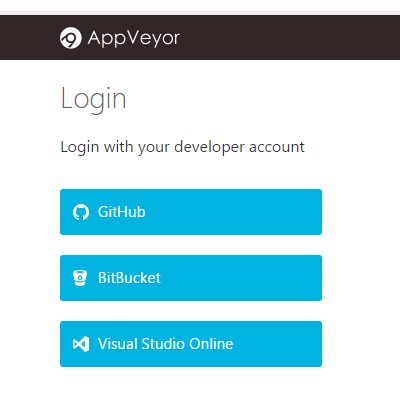


Рисунок 4.9 Окно авторизации

Добавим новый проект:

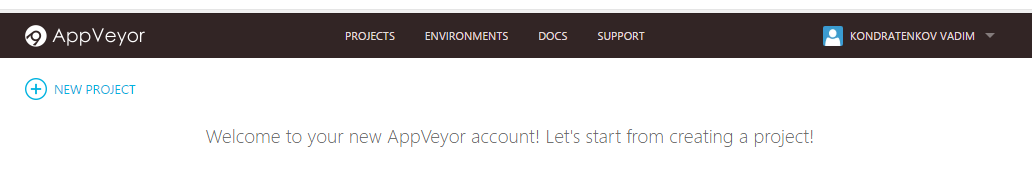


Рисунок 4.10 Окно проектов

Выберем требуемый репозиторий (KOMK\_Asp\_ServiseDesk):

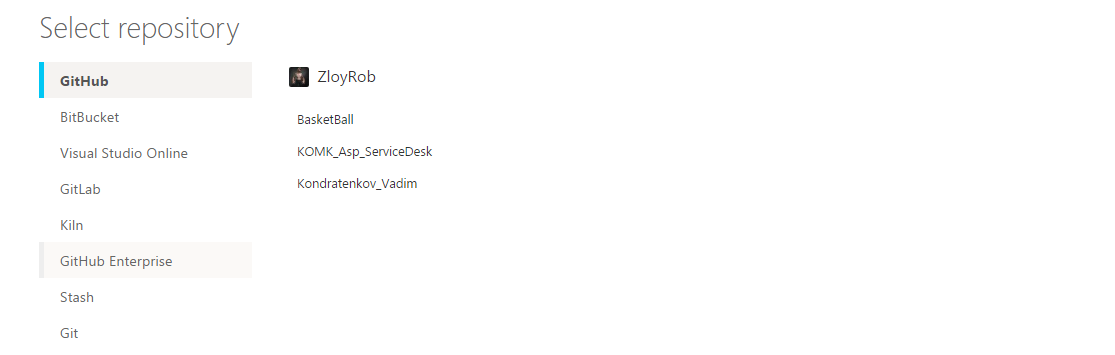


Рисунок 4.11 Выбор репозитория

Создадим в локальном репозитории файл конфигурации *appveyour.yml* и загрузим его:

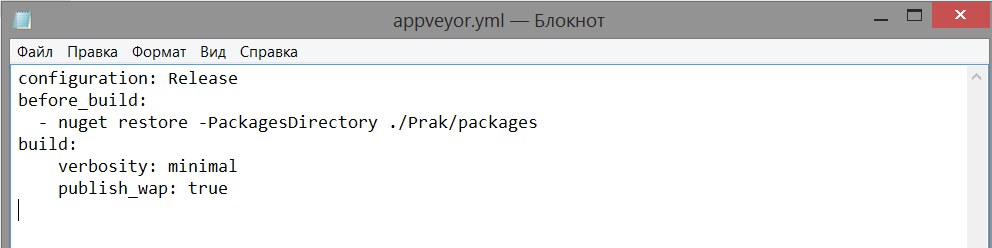


Рисунок 4.12 файл конфигурации appveyour.yml

Теперь на сайте <https://ci.appveyor.com/> мы можем наблюдать за ходом сборки и тестирования нашего приложения.

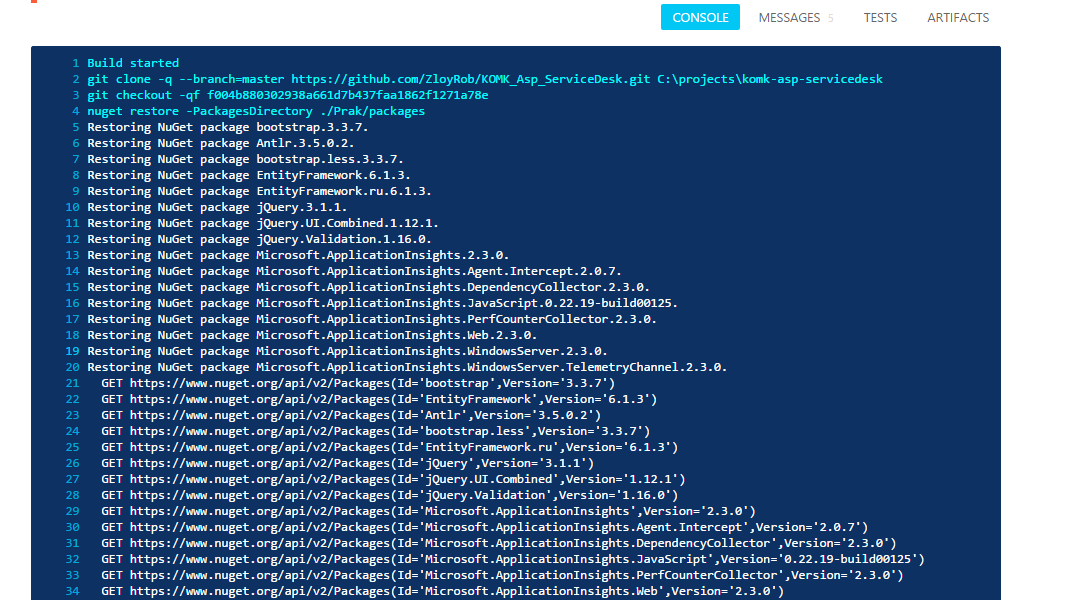


Рисунок 4.13 Сборка решения.

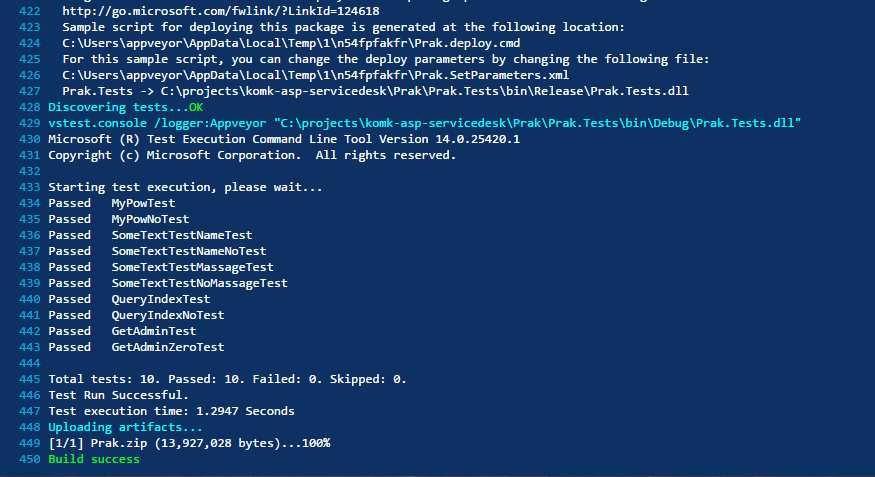


Рисунок 4.14 Ход тестирования.

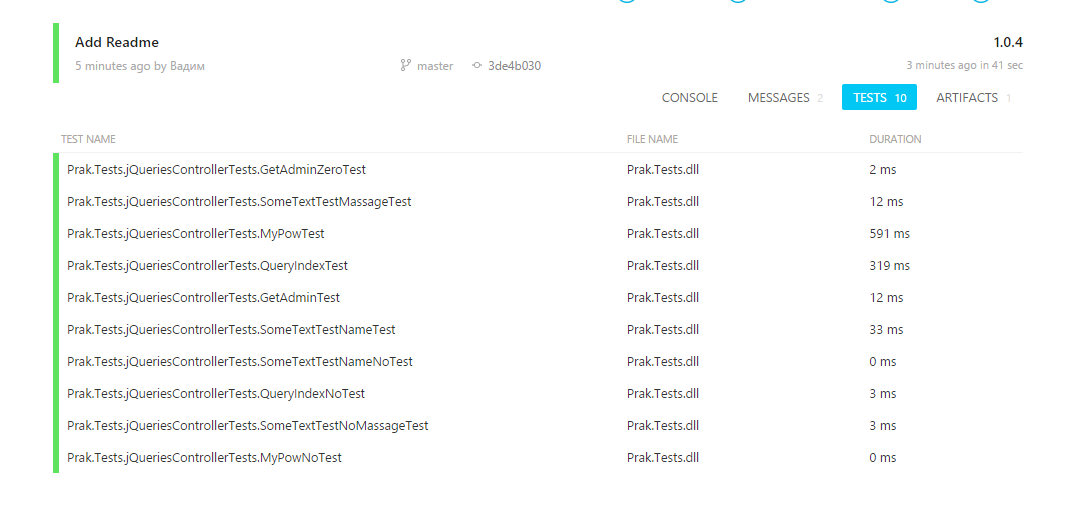


Рисунок 4.15 Отчет о пройденных тестах

В случае если какой-либо тест не прошел сервис отобразит проваленные тесты (рисунок 3.16).

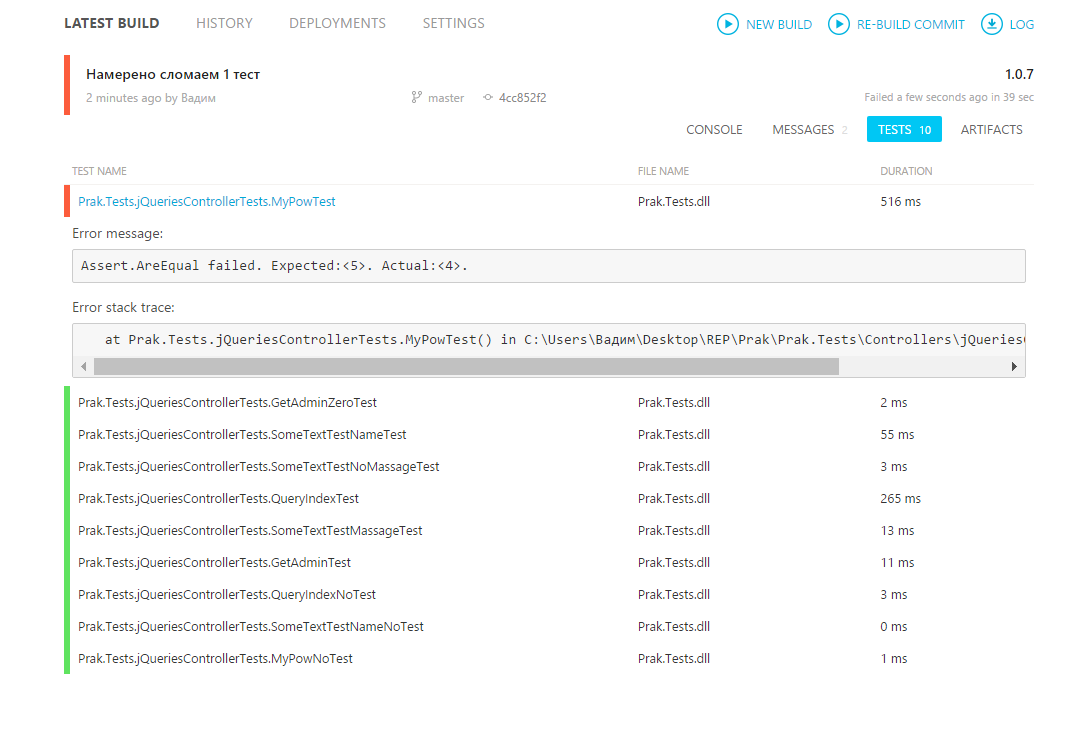


Рисунок 4.16 Отчет о проваленных тестах

# Технологии разработки и программная реализация

## Выбор операционной системы

В качестве операционной системы для развертывания информационной системы выбираем Windows 7.

Выбор обусловлен следующими факторами:

* На машинах предприятия, для которого предназначена разрабатываемая информационная система, установлен Windows 7 и комплект необходимых для работы программ, смена операционной системы будет нецелесообразной;
* Система обладает достаточно высоким уровнем надежности;
* Система обладает достаточно высоким уровнем безопасности;
* Система является удобной и простой в использовании.

## Выбор типа взаимодействия пользователя с системой, технологии пользовательского интерфейса

Для взаимодействия пользователя с системой создаем интерфейс в виде web приложения на основе технологии ASP.NET, размещаемого на сервере.

Данный выбор обусловлен низкой производительностью рабочих станций пользователей КОМК, ввиду этого было решено положить основную нагрузку на сервер, реализовав «Тонкого клиента». Благодаря тому, что приложение не требуется запускать с рабочей станции, а лишь получать содержимое страниц с сервера, скорость выполнения базового процесса будет выше.

## Выбор технологии взаимодействия пользовательских компонент с данными

Взаимодействие приложения с данными будет построено с использованием технологии ADO.NET Entity Framework.

Данный выбор обусловлен тем, что разработка приложения ведется с использованием технологии из стека Microsoft ASP.NET. Выбор технологии из

того же стека сократит время на разработку приложения, ввиду снижения временных затрат на налаживание взаимодействия между технологиями.

## Выбор языка и среды программирования

В качестве среды программирования для системы выбираем Microsoft Visual Studio, а в качестве языка C#. Данный язык и среда являются универсальными инструментами программирования, поэтому они подходят для решения поставленной задачи по созданию ИС.

Данный выбор обусловлен проектными требованиями к приложению. Приложение должно быть разработано на языке C#.

## Определение состава компонент

При формировании компонент рекомендует ориентироваться по трем логическим сервисным уровням:

* пользовательскому;
* прикладному;
* уровню данных.

**Пользовательский уровень** — отображает данные и позволяет пользователю редактировать их.

**Прикладной уровень** — здесь реализованы бизнес-правила и ограничения на данные. И хотя его сервисы используются презентационным уровнем (пользовательским), он не привязан к какому-либо клиенту — сервисы прикладного уровня доступны любому клиенту. Бизнес-правила выражаются в форме прикладных алгоритмов, корпоративных правил и т. д. Бизнес правила обычно реализуются отдельным модулем на централизованном сервере, что дает возможность доступа к нему сразу нескольким клиентам.

**Уровень данных** — прикладной уровень не знает, как и где хранится обрабатываемая им информация. В этом вопросе он полагается на сервисы доступа к данным, выполняющие всю работу по получению и передаче данных. Сервисы доступа к данным также реализуются в виде изолированных модулей, «знающих» о месте хранения информации. Таким образом, если хранилище перемещено или изменен его формат, потребуется обновить только сервисы доступа к данным. Каждый модуль доступа к данным, как правило, отвечает и за целостность хранилища (например, реляционной базы данных). Для многоуровневых приложений в качестве хранилища информации подходят простые системы управления базами данных (СУБД), необходимые для обслуживания данных в таблицах и быстрой выборки информации (например, с помощью индексов).

Разрабатываемая информационная система будет построена по трехуровневой технологии – пользовательский уровень, прикладной уровень и уровень данных.

## Разработка схемы развертывания

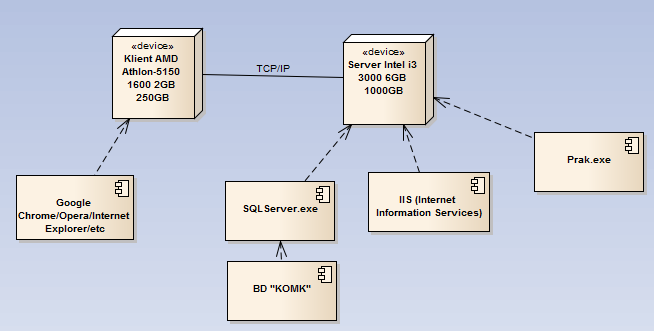
Диаграмма среды развертывания представлена на рисунке 5.1

Рисунок 5.1 Среда развертывания

На серверной машине предприятия уже установлен сервер SqlServer. На сервер необходимо развернуть базу данных KOMK, установить IIS сервер, а так же загрузить разработанное приложение.

На клиентских машинах необходимо наличие браузера.

После загрузки всех компонентов необходимо провести тестовый запуск программы.

## Требования к системе

При разработке информационной системы были выявлены следующие требования:

* Возможность составления заявок;
* Возможность обработки заявок сотрудником тех. поддержки;
* Возможность добавления работ к заявке;
* Возможность смены статуса работы;
* Возможность добавления комментария к работе;
* Возможность смены статуса заявки;
* Возможность назначения исполнителей на работы.

## Разработка компонент

* Главная страница приложения представлена на рисунке 5.2. Реализованы возможности создания заявки, работы, смены статуса заявки, работы. Любые действия с заявками и работами возможны только после авторизации пользователя.

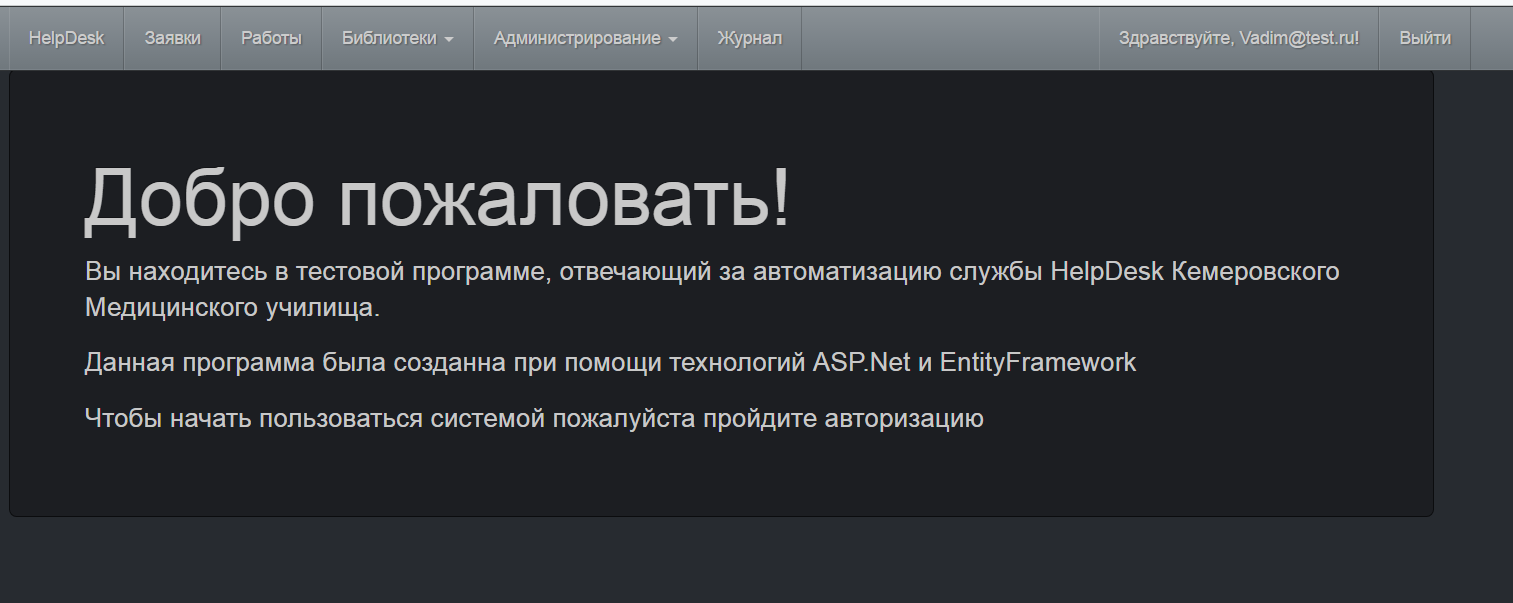


Рисунок 5.2. Главная страница приложения

* Страница создания заявки представлена на рисунке 5.3. Реализованы возможность добавления заявки.

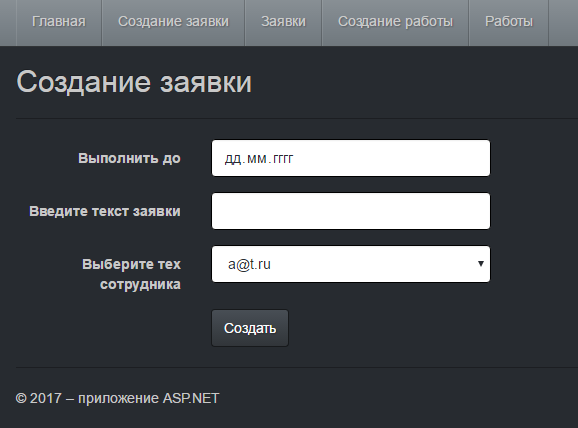


Рисунок 5.3. Страница создания заявки

* Страница просмотра всех заявок представлена на рисунке 5.4, реализована возможность смены статуса заявки.

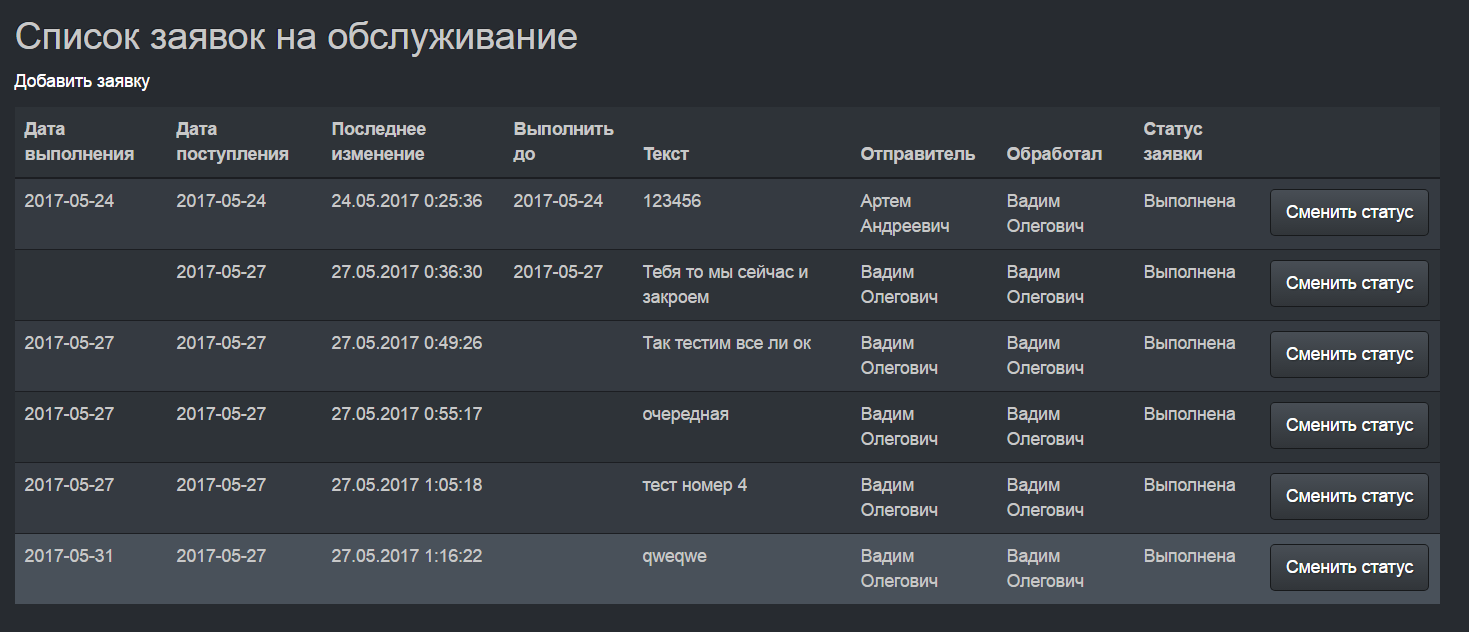


Рисунок 5.4. Страница просмотра всех заявок

* Страница смены статуса заявки представлена на рисунке 5.5. Реализована возможность смены статуса заявки.

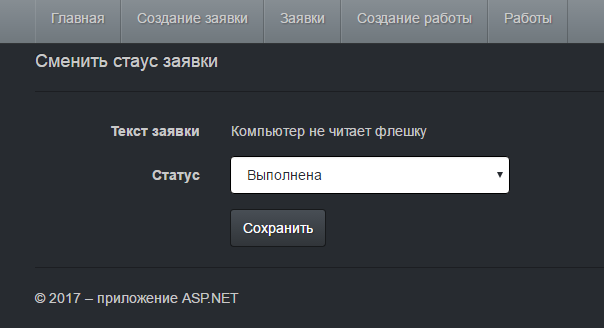


Рисунок 5.5. Страница смены статуса заявки

* Страница создания работы представлена на рисунке 5.6. Реализована возможность создания работы.

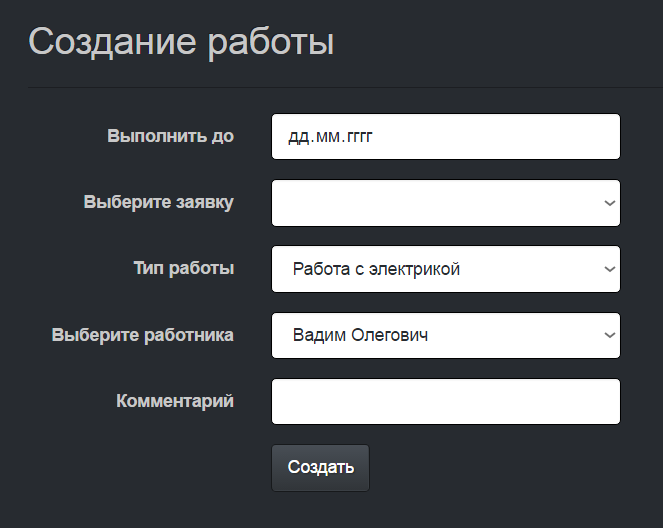


Рисунок 5.6. Страница создания работы

* Страница просмотра работ представлена на рисунке 5.7. Реализованы возможности смены статуса работы, смены исполнителя, добавки комментария.

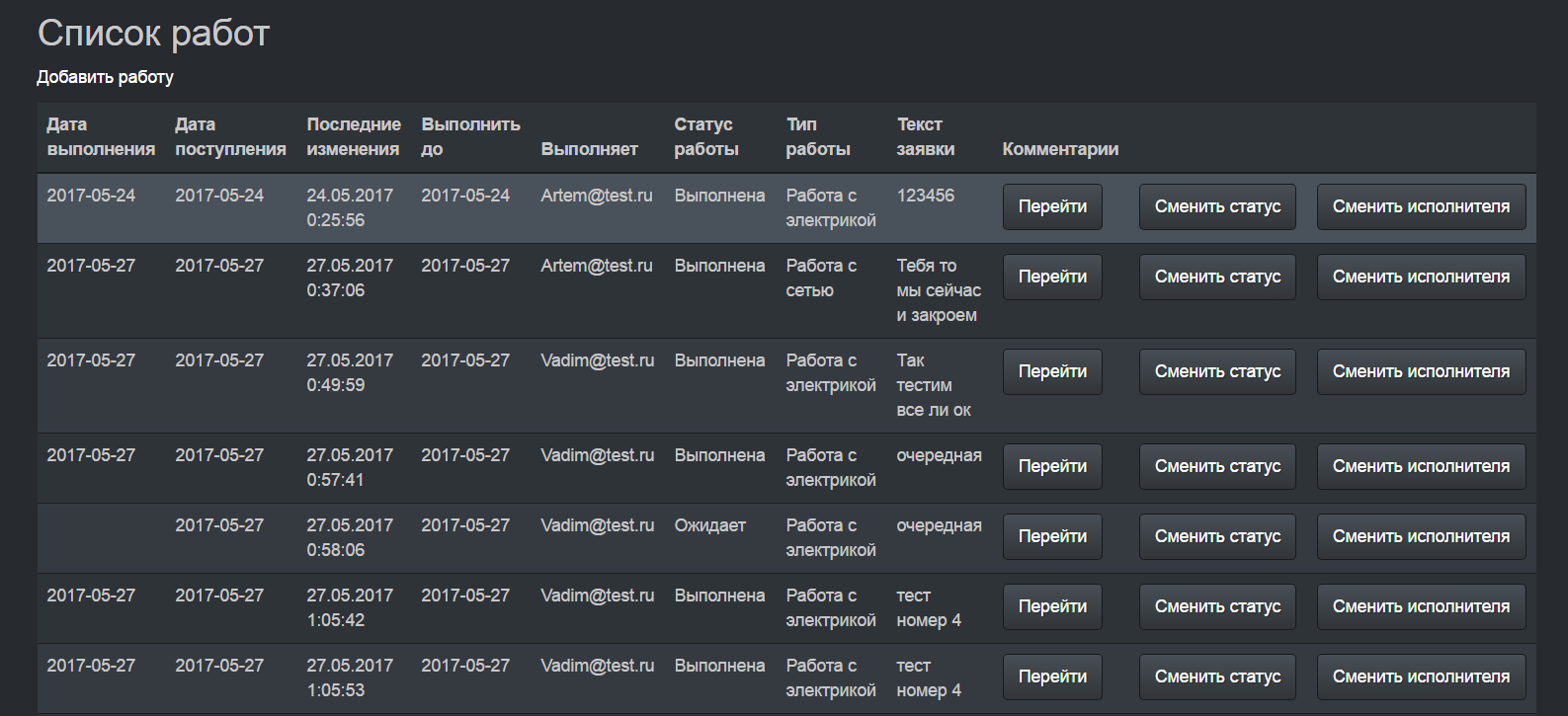


Рисунок 5.7. Страница просмотра работ.

* Страница смены статуса работы представлена на рисунке 5.8. Реализована возможность смены статуса работы.

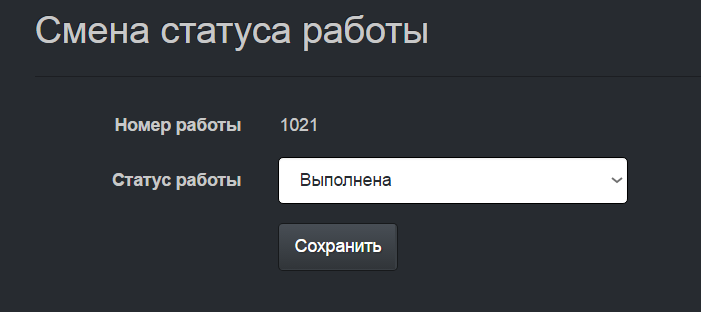


Рисунок 5.8. Страница смены статуса работы.

* Страница смены исполнителя представлена на рисунке 5.9. Реализована возможность смены исполнителя на работе.

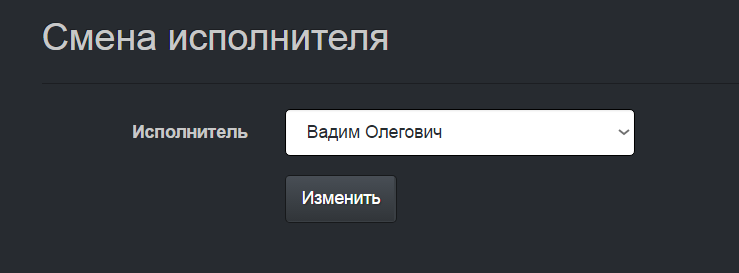


Рисунок 5.9. Страница исполнителя.

* Страница авторизации пользователя представлена на рисунке 5.10. Реализована возможность авторизации пользователя.

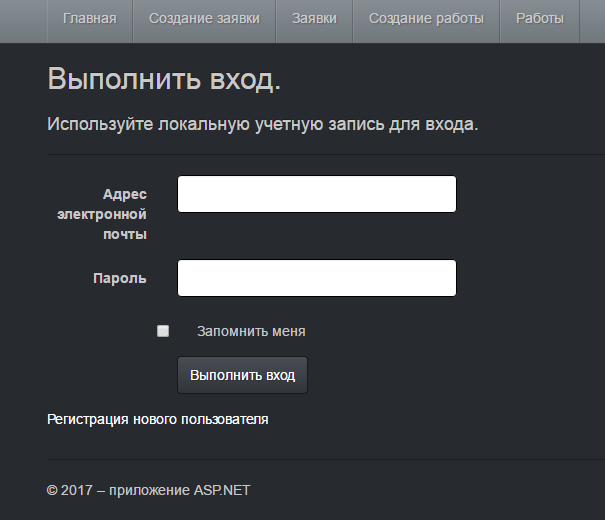


Рисунок 5.10. Страница авторизации пользователя

* Страница регистрации пользователя представлена на рисунке 5.11. Реализована возможность регистрации пользователя.

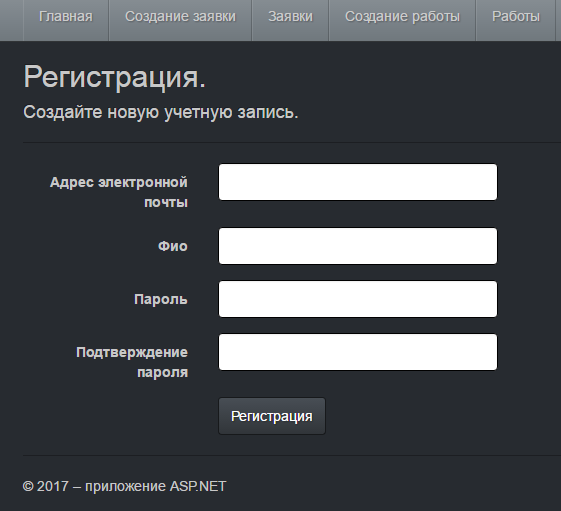


Рисунок 5.11. Страница регистрации пользователя

* Страница «Журнал событий» представлена на рисунке 5.12, на данной странице отображаются действия пользователей в системе.

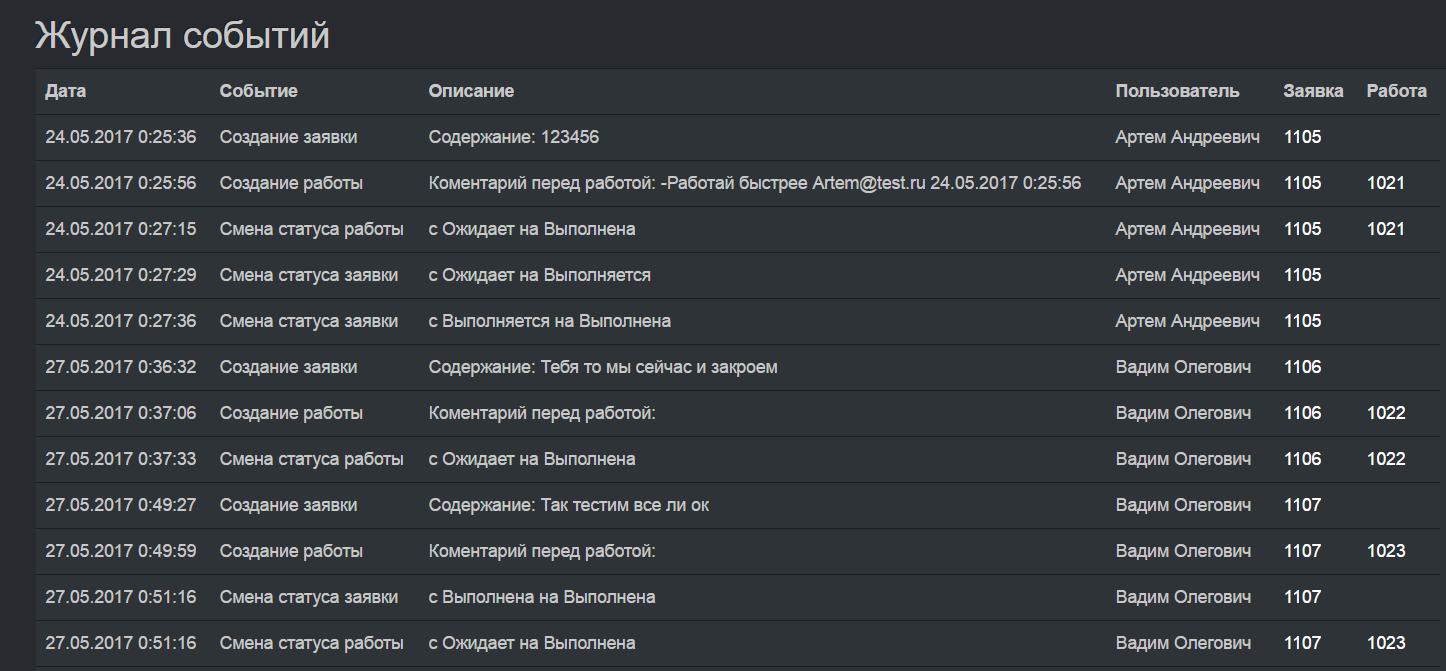


Рисунок 5.12. Страница «Журнал событий»

Блок «Библиотеки» (доступен только сотруднику тех. поддержки/администратору) содержит следующие страницы:

* Страница «События», отображает события, которые заносятся в журнал. Реализована возможность просмотра;

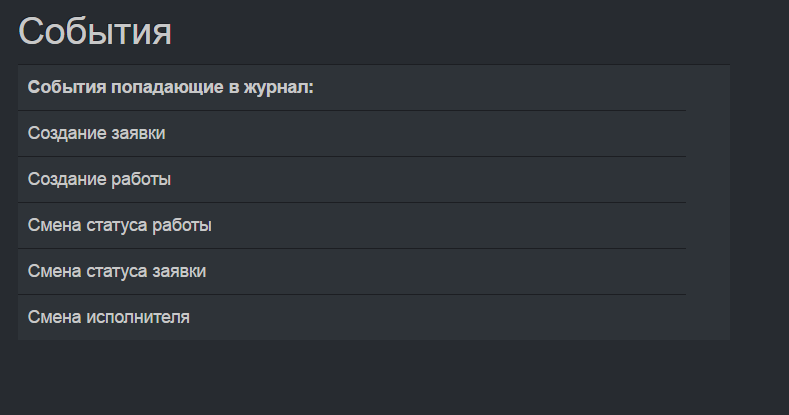


Рисунок 5.13. Страница «События»

* Страница «Конфигурирование статусов заявок», отображает статусы заявок, фигурирующие в системе. Реализована возможность создания, изменения и удаления статусов;

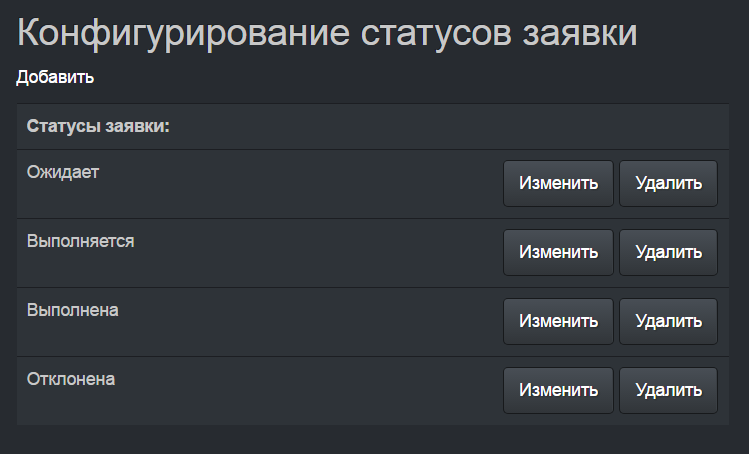


Рисунок 5.14. Страница «Конфигурирование статусов заявок»

* Страница «Конфигурирование статусов работ», отображает статусы работ, фигурирующие в системе. Реализована возможность создания, изменения и удаления статусов;

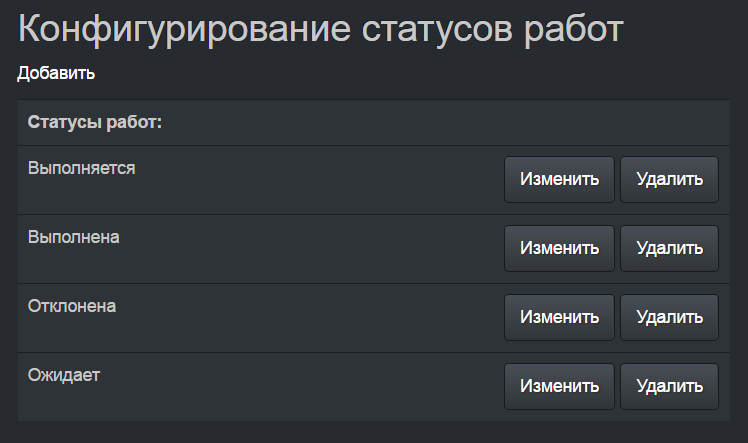


Рисунок 5.15. Страница «Конфигурирование статусов работ»

* Страница «Конфигурирование типов работ», отображает типы работ, фигурирующие в системе, и соотносит им умения необходимые для их выполнения. Реализована возможность создания, изменения и удаления работ;

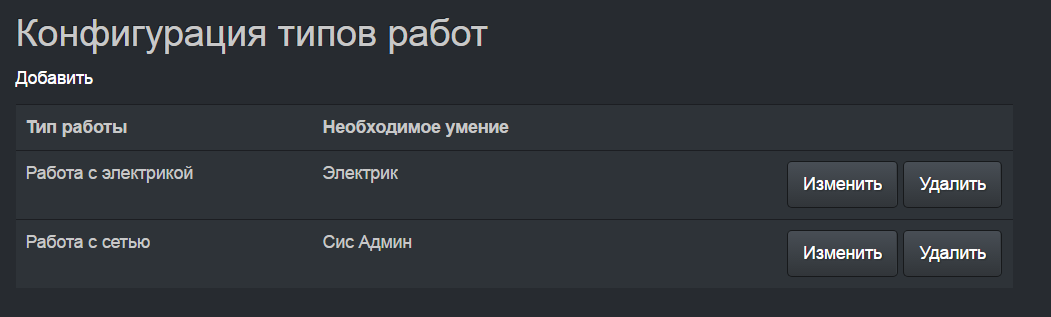


Рисунок 5.16. Страница «Конфигурирование типов работ»

* Страница «Конфигурирование умений», отображает умения пользователей фигурирующие в системе. Реализована возможность создания, изменения и удаления умений;

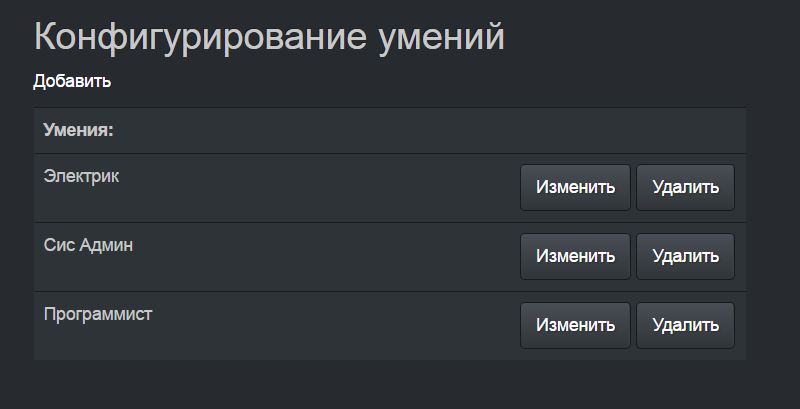


Рисунок 5.17. Страница «Конфигурирование умений»

Блок «Администрирование» (доступен только сотруднику тех. поддержки/администратору) содержит следующие страницы:

* Страница «Пользователи», содержит информацию о пользователях, зарегистрированных в системе, реализована возможность создания, удаления и изменения пользователей. Также реализована возможность импорта/экспорта пользователей в Excel файл;

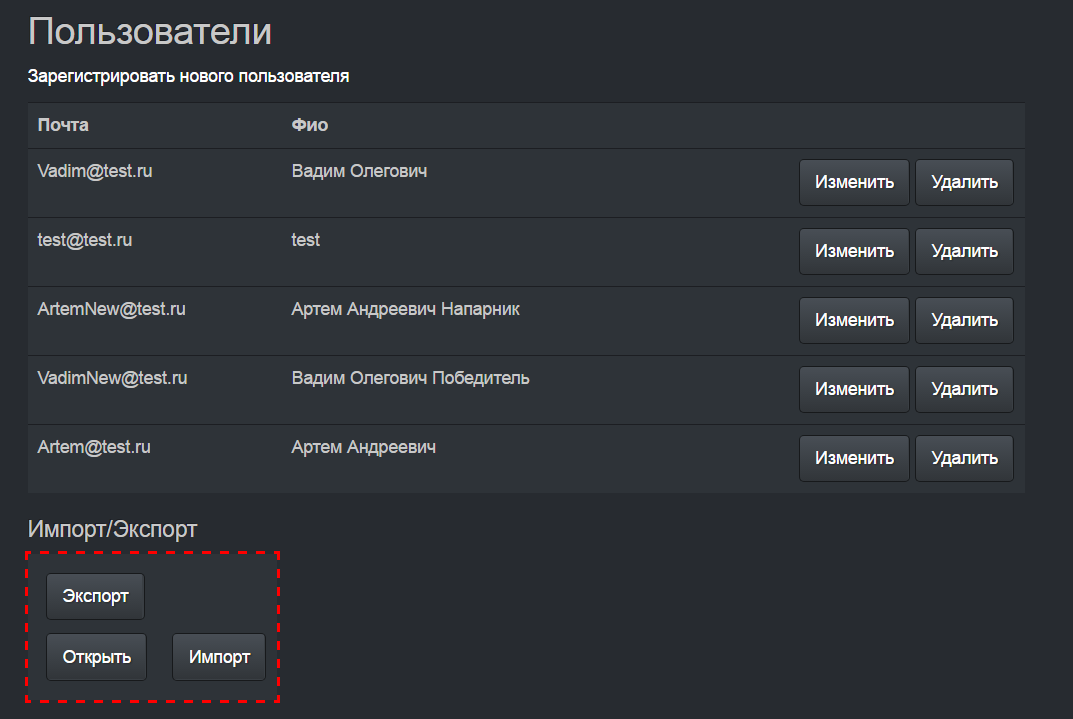


Рисунок 5.18. Страница «Пользователи»

* Страница «Распределение умений», содержит информацию об умениях пользователей, реализована возможность назначения пользователям новых умений, так же есть возможность изменения и удаления умений;

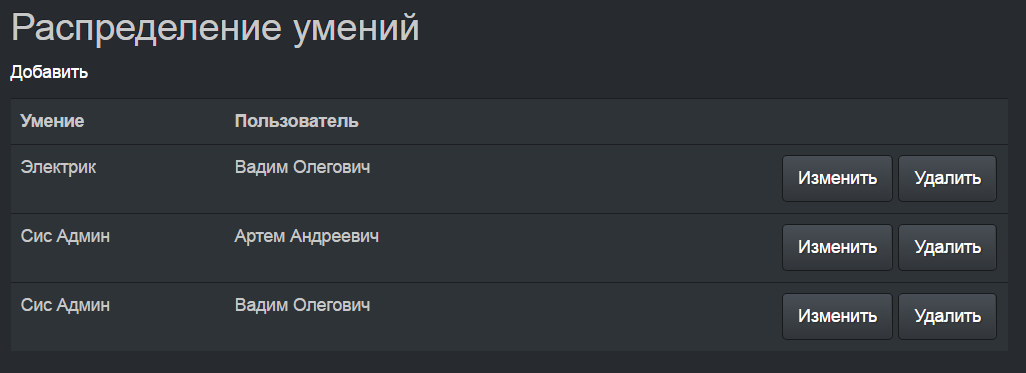


Рисунок 5.19. Страница «Распределение умений»

* Страница «Распределение ролей», содержит информацию о ролях пользователей (роль пользователя определяет его уровень доступа к функционалу системы), реализована возможность смены роли пользователя.

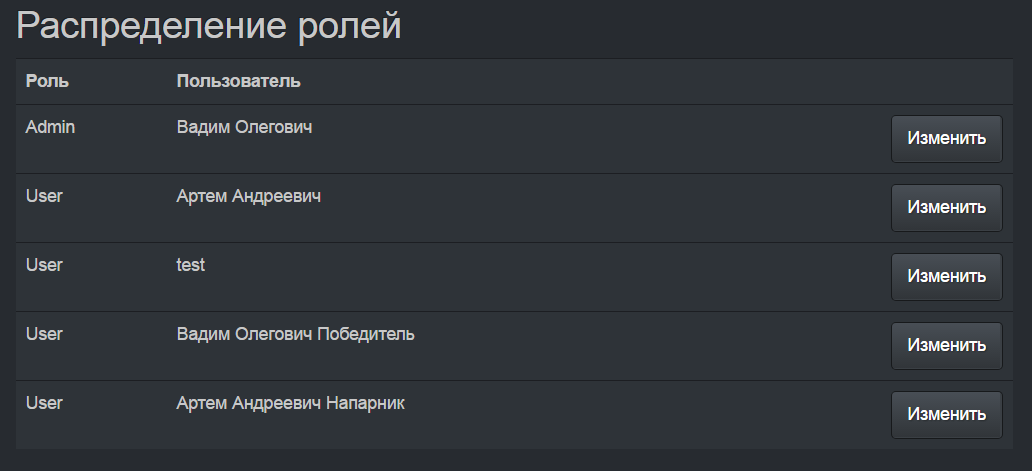


Рисунок 5.20. Страница «Распределение ролей»

Связь между страницами приложения представлена на рисунке 5.21.

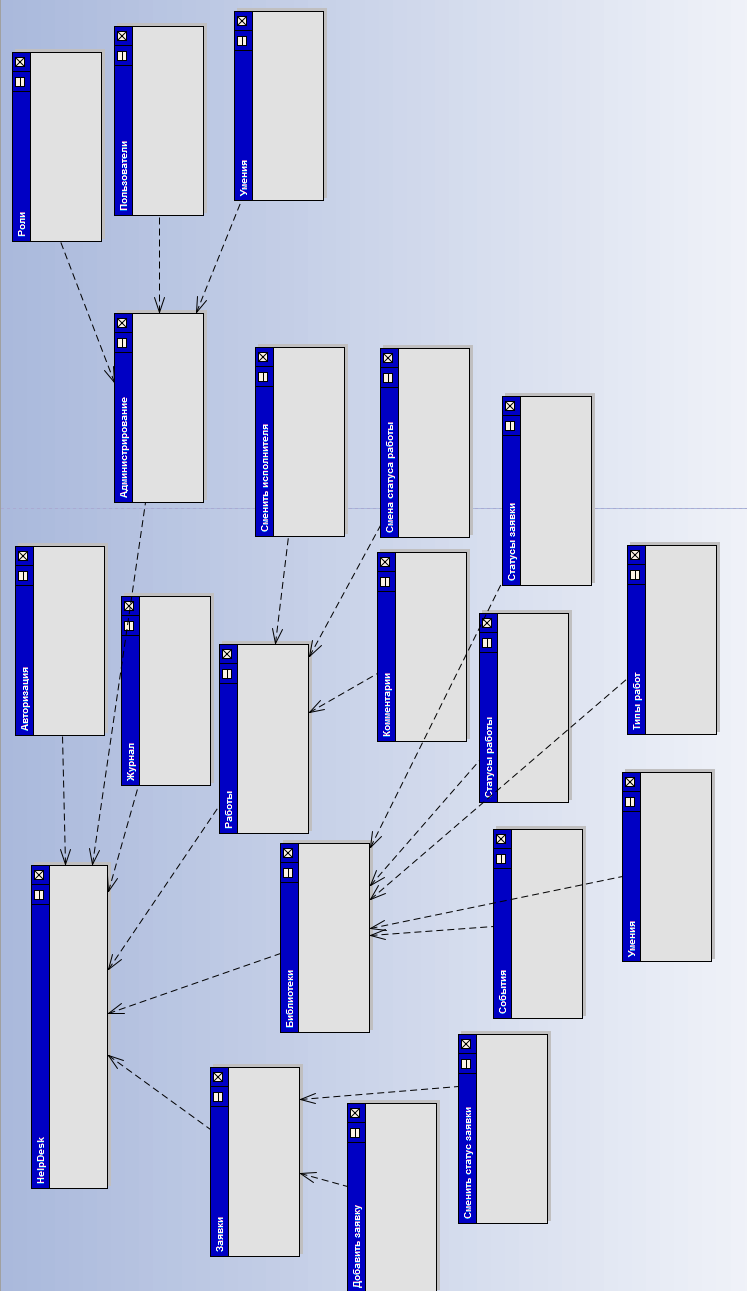


Рисунок 5.21. Связь между страницами приложения.

## Характеристика разработанного решения

Приложение было разработано на основе технологий ADO.NET Entity Framework, с использованием среды разработки Visual Studio 2015 и СУБД MS SQL Server 2012. Была выполнена реализация архитектурных вариантов использования и связанных с ними отношениями включения вариантов использования:

* Создание заявки;
* Работа с заявкой;
  + Составление списка работ по заявке;
  + Назначение исполнителя на выполнение работы;
  + Смена статуса заявки;
* Конфигурирование системы;
  + Формирование списка типов работ;
  + Формирование списка умений пользователей;
  + Назначение прав пользователей;
* Смена статуса работы;
  + Добавление комментариев к работе;
* Журналирование действий пользователей в системе.

Состав компонентов, содержащих программный код представлен на рисунке 36, состав экранных форм представлен на рисунке 5.22.

Рисунок 5.22. Состав компонентов программного кода.

## Разработка вопросов тестирования

Для тестирования выбраны следующие сценарии «Создание заявки», «Создание работы», «Закрытие заявки», «Смена статуса работы».

Сценарий «Создание заявки»:

1. Пользователь авторизуется.

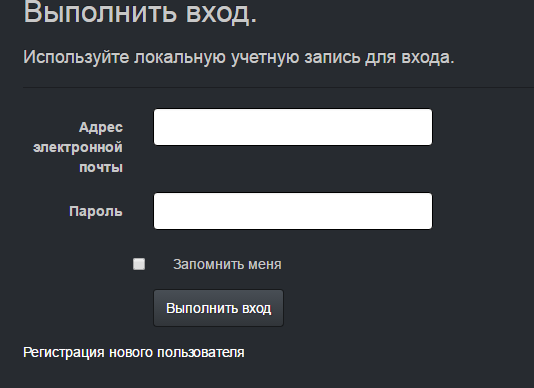


Рисунок 5.23. Вкладка «Авторизация»

1. Открывает вкладку «Создание заявки».

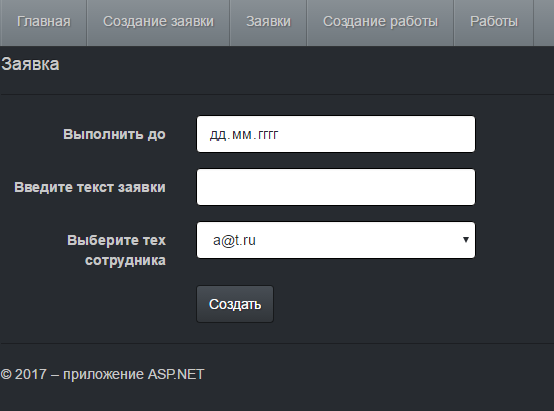


Рисунок 5.24. Вкладка «Создание заявки»

1. На вкладке «Создание заявки» необходимо заполнить поля данных заявки (срок выполнения, текст заявки, какому сотруднику послать заявку). После нажатия кнопки «Создать» в таблицу базы данных jQuery добавляется запись о созданной заявке, а на вкладке «Заявки» обновляется список заявок.

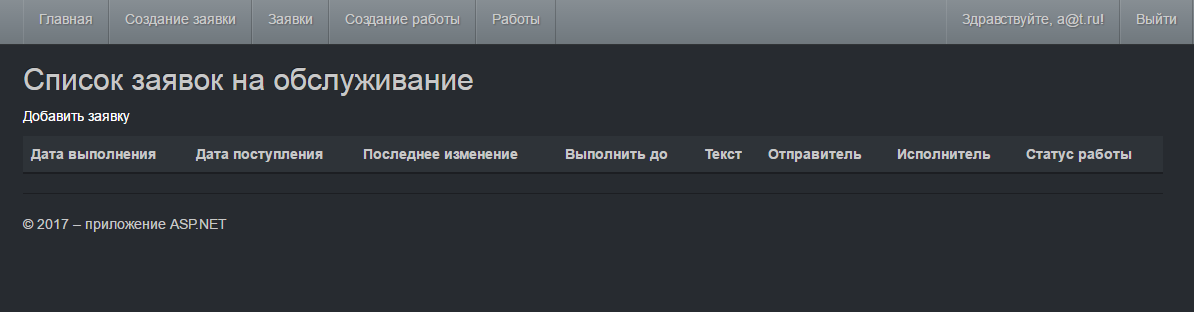


Рисунок 5.25. Вкладка «Заявки»

Сценарий «Создание работы»:

1. Пользователь авторизуется.

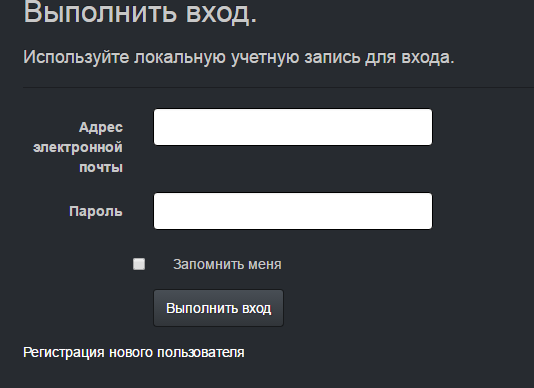


Рисунок 5.26. Вкладка «Авторизация»

1. Открывает вкладку «Создание работы».

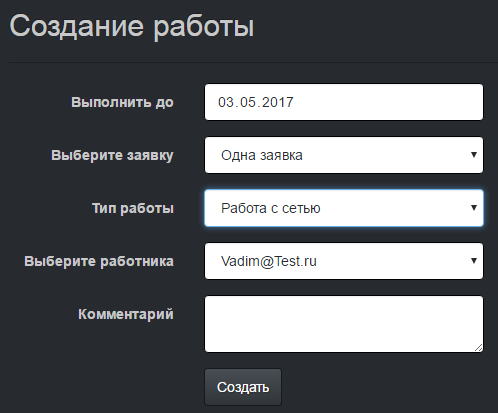


Рисунок 5.27. Вкладка «Создание работы»

1. На вкладке «Создание работы» необходимо заполнить поля данных заявки (срок выполнения, заявка к которой относится работа, тип работы, исполнитель). После нажатия кнопки «Создать» в таблицу базы данных jWorkList добавляется запись о созданной работе, а на вкладке «Работы» обновляется список работ.



Рисунок 5.28. Вкладка «Работы»

Сценарий «Закрытие заявки»:

1. Пользователь авторизуется.

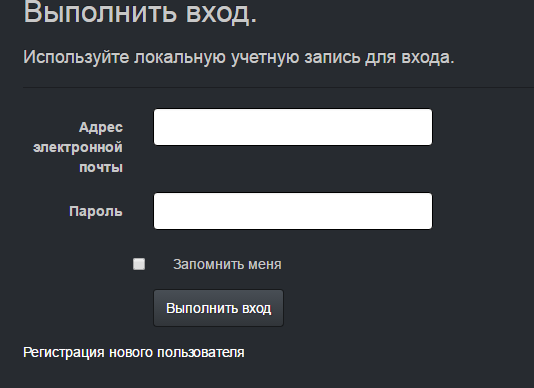


Рисунок 5.29. Вкладка «Авторизация»

1. Переходит на вкладку «Заявка».

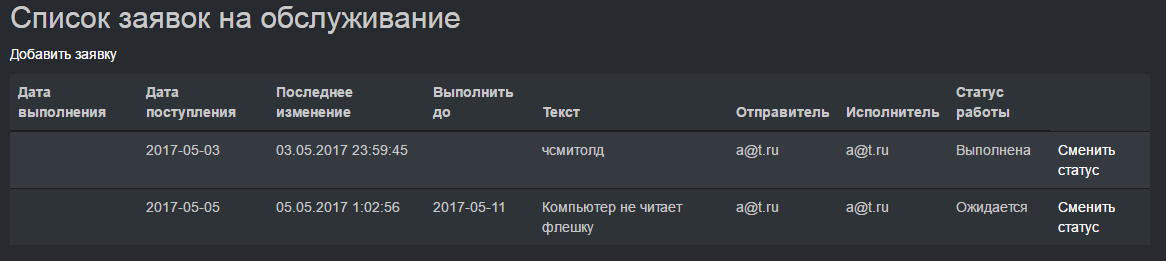


Рисунок 5.30. Вкладка «Заявки»

1. Выбирает требуемую заявку и переходит по ссылке «Сменить статус». На открывшейся вкладке меняют статус заявки на “выполнена” и сохраняет статус.

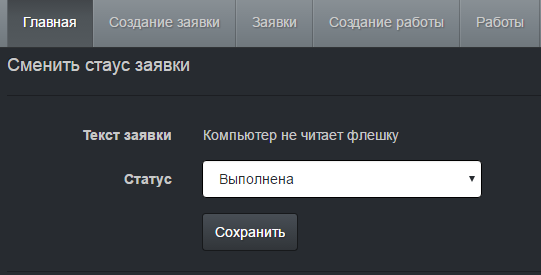


Рисунок 5.31. Вкладка «Смена статуса заявки»

1. На вкладке «Заявка» должен обновится статус заявки.

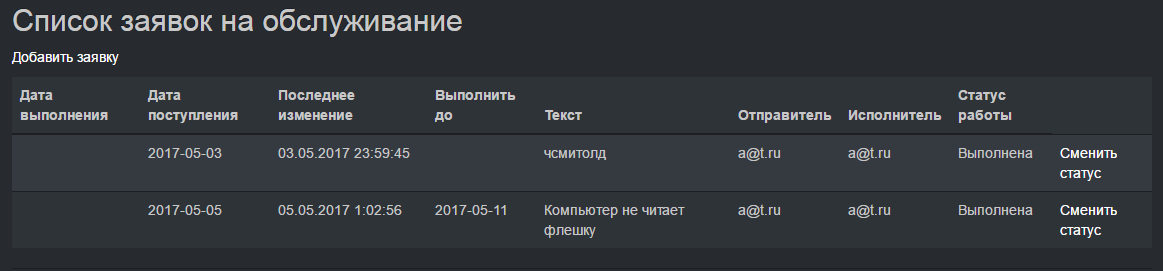


Рисунок 5.32. Вкладка «Заявки»

Сценарий «Смена статуса работы»:

1. Пользователь авторизуется.

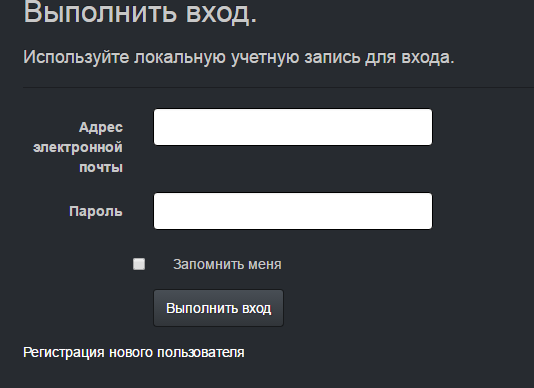


Рисунок 5.33. Вкладка «Авторизация»

1. Переходит на вкладку «Работы».



Рисунок 5.34. Вкладка «Работы»

1. Выбирает работу, статус которой надо сменить и переходит по ссылке «Сменить статус». На открывшейся вкладке меняют статус работы, при желании пользователь может оставить комментарий к работе. Сделанные изменения сохраняются.

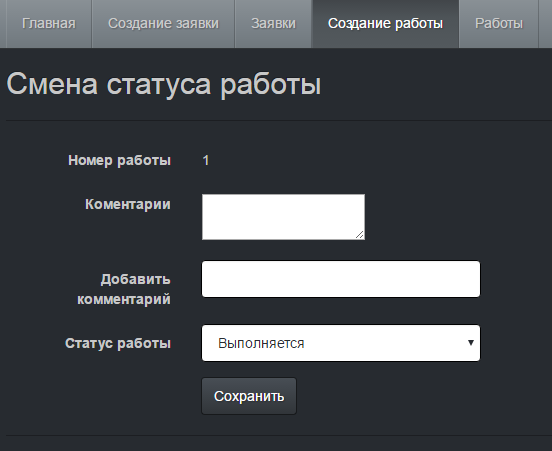


Рисунок 5.35. Вкладка «Смена статуса работы»

# Аппаратная и административная интеграция ИС

## Разработка схемы развертывания

Чтобы развернуть разработанное приложение на предприятии, необходимо установить приложение на сервер, затем развернуть на нем Backup прилагаемой БД “Coursach.bacpak”, требуется наличие IIS сервера. Установка MS SQL Server 2012 и службы IIS на сервер не требуется, т.к. они уже установлены на предприятии. Для работы с приложением на рабочей станции достаточно перейти на определенную веб-страницу с любого браузера.

Схема развертывания представлена на рисунке 6.1.



Рисунок 6.1. Диаграмма развертывания ИС.

## Формулировка требований к физическим устройствам и сетевому оборудованию, состав рабочих мест

Данная система была разработана для последующей загрузки на сервер с установленной ОС Windows Server.

Для функционирования разработанной ИС необходимо следующее оборудование:

* Сервер СУБД - для размещения БД;
* Рабочие станции – будут использоваться для доступа к клиентскому приложению;
* Коммутатор – для соединения нескольких узлов локальной сети;
* Сетевой кабель, для объединения рабочих станций в единую локальную вычислительную сеть (ЛВС).

Для работы с единой базой данных необходимо подключить все рабочие места к сети.

С системой будут работать как минимум 4 рабочие станции, которые должны быть подключены к сети.

Сервер будет располагаться в специально отведенной для него комнате.

Для работоспособности системы будет использоваться сеть с технологией передачи данных Fast Ethernet 100Base-TX с максимальной скоростью передачи данных 100 Мбит\сек.

## Выбор состава аппаратных средств

В качестве сервера рекомендуется использовать компьютер со следующими характеристиками:

* CPU: Intel core i3 3000;
* RAM: 6GB;
* HDD: от 1000 GB;
* ОС Windows Server.

В качестве рабочих станций рекомендуется использовать компьютер со следующими характеристиками:

* CPU: AMD Athlon-5150 1600;
* RAM: 2GB;
* HDD: 250GB;
* OC Windows XP / Windows 7.

## Расчет потребности персонала.

Расчет потребности персонала производится с учетом требований, предъявляемым к системе.

Для обеспечения безопасности и надежности информационной системы необходим следующий персонал:

* **Главный повар.**

Для обеспечения возможности генерирования меню требуется должность повара. Эта должность необходима для использования функций автоматической генерации меню.

* **Заведующий склада.**

Для ввода данных о поставках и перерасчёта стоимости ингредиентов.

* **Менеджер.**

Для добавления новых блюд, ингредиентов. Он изменяет состав блюд, может создавать новые типы блюд и перерасчитывает стоимость блюд.

* **Администратор**

Для обеспечения доступа сотрудников к ресурсам сайта. Он назначает пользователям роли.

На предприятии имеются все необходимые специалисты. Вопрос найма новых специалистов рассматривать не нужно.

## Выбор сетевой архитектуры и технологии

Выбор архитектуры

На предприятии уже настроена ЛВС, со следующими параметрами, представленными в таблице 11.

Таблица 11 Основные характеристики сети.

|  |  |
| --- | --- |
| Компонент/характеристика | Вариант сети |
| Топология | Звезда |
| Линия связи | Экранированная витая пара cat 5е. |
| Сетевые адаптеры | Fast Ethernet |
| Ретрансляторы (повторители, концентраторы, коммутаторы, мосты, маршрутизаторы, шлюзы) | Коммутаторы (D-Link, Cisco), роутер (Cisco). |
| Управление совместным использование ресурсов | Клиент - серверная архитектура построения. |
| Поддерживаемые приложения | Электронная почта, работа с БД. |

Схема ЛВС представлена в приложении 1.

Выбор технологии и аппаратных средств. Расчет сети

На предприятие установлена стандартная сеть на базе технологии Fast Ethernet.

Поскольку сеть уже создана, а дальнейшее расширение сети не связано напрямую с разрабатываемой системой, то стоимость сети в проект не закладывается.

Таблица 12 Расчет сети.

|  |  |  |  |
| --- | --- | --- | --- |
| Наименование | Цена, руб. | Количество | Стоимость, руб |
| Оборудование: | | | |
| Кабель UTP 5e, м | 12 | 80 | 960 |
| Коннектор RJ-45, шт | 4 | 10 | 40 |
| Кабель-канал, м | 10 | 80 | 800 |
| Монтаж: | | | |
| Монтаж кабель-канала, м | 25 | 80 | 2000 |
| Обжим коннектора RJ-45, шт | 23 | 10 | 230 |
| Итого:4030 | | | |

# Общие вопросы администрирования

## Определение стратегии администрирования на уровне руководства.

В ИС реализуются следующие цели администрирования:

* обеспечение безопасности и целостности данных путем управления доступом пользователей (определение каждого пользователя базы данных, назначение паролей):
* обеспечение бесперебойной работы информационной системы колледжа;
* внедрение инструкций для пользователей и администраторов.

## Определение объектов администрирования на уровне предприятия.

Объектами администрирования на уровне предприятия являются:

* рабочие станции;
* сетевые устройства (маршрутизатор);
* база данных и её объекты (таблицы БД, база выполняется на основе MS SQL Server 2012);
* программное обеспечение, в том числе разрабатываемая ИС;

## Политика администрирования на уровне предприятия.

* Для фильтрации трафика поступающего из Интернета на пограничном маршрутизаторе настроен брандмауэр;
* Разграничение по ролям осуществляется администратором с использованием службы Active Directory;
* Ограничение трафика проходящего по сети производится при помощи разделения рабочих станций предприятия по VLAN-ам;
* Конфигурирование новых рабочих станций производится с использованием Active Directory;
* Режим работы сети – круглосуточный. Основным транспортным протоколом для передачи данных является протокол TCP/IP;

## Определение объектов администрирования на уровне разрабатываемой ИС.

Объекты администрирования на уровне разрабатываемой ИС:

В процессе анализа были выделены следующие объекты администрирования на уровне разрабатываемой ИС:

* разработанное приложение (клиентская часть, таблицы БД);
* база данных, включающая таблицы:
  + «AspNetUsers» - содержит данные о пользователях;
  + «Dish» - содержит данные о существующих блюдах;
  + «Ingredient» - содержит данные о существующих ингредиентах;
  + «Menu» - содержит данные о существующих меню;
  + «DishTypes» - содержит данные о существующих типах блюд;
  + «DishComposition» - содержит данные о составе конкретных блюд;
  + «MenuComposition» - содержит данные о составе конкретных меню;
  + «AspNetUserRoles» - содержит данные о ролях конкретных пользователей;
  + «MenuRequirement» - содержит данные о требованиях к конкретным меню;
  + «Entrance» - содержит данные о поставках;
  + «EntranceComposition» - содержит данные о составе конкретных поставок.
* сервер.

## Политика администрирования на уровне разрабатываемой ИС.

В разрабатываемой системе присутствует разделение по ролям:

* Сотрудник кухни
  + Имеет полный доступ к таблицам БД «Menu», «MenuComposition», «MenuRequirement», содержащим информацию о меню. Способен создавать (с использованием функционала разработанной ИС) новые меню.
  + Способен создавать новые меню, созданные меню записываются в таблицу БД «Menu». Создание меню производится с использованием функционала приложения.
  + Способен создавать новые требования к меню, созданные требования записываются в таблицу БД «MenuRequirement». Создание требований производится с использованием функционала приложения.
  + Способен создавать новые записи в таблице «MenuComposition», задает только требования в таблице «MenuRequirements» и таблица «MenuComposition» заполняется без его участия функционалом приложения.
* Сотрудник склада
  + Имеет полный доступ к таблицам БД «Entrance» и «EntranceComposition», содержащим информацию о поставках.
  + Имеет частичный доступ к таблице БД «Ingredient», содержащей информацию о ингредиентах.
  + Способен создавать новые поставки, созданные поставки записываются в таблицу БД «Entrance». Создание поставок производится с использованием функционала приложения.
  + Способен создавать состав поставки, созданные составы поставки записываются в таблицу БД «EntranceComposition». Создание поставок производится с использованием функционала приложения
  + Способен менять стоимость ингредиентов на складе, которые были включены в поставку. Изменение ингредиента производится с использованием функционала приложения.
* Менеджер
  + Имеет полный доступ к таблицам БД «Dish», «Ingredient», «DishTypes», «DishComposition».
  + Способен создавать новые типы блюд, созданные типы блюд записываются в таблицу БД «DishTypes». Создание типов блюд производится с использованием функционала приложения.
  + Способен создавать новые блюда, созданные блюда записываются в таблицу БД «Dish». Создание блюд производится с использованием функционала приложения.
  + Способен создавать новые ингредиенты, созданные ингредиенты записываются в таблицу БД «Ingredient». Создание ингредиентов производится с использованием функционала приложения.
  + Способен создавать состав блюд, созданный состав блюда записываются в таблицу БД «DishComposition». Создание состава блюд производится с использованием функционала приложения.
  + Обладает возможностью конфигурирования системы, изменяя содержимое таблицы БД «Dish».
* Администратор
  + Имеет полный доступ к таблице БД «AspNetUserRoles», содержащей информацию о правах пользователя.
  + Способен разрешать и запрещать пользователям доступ к ресурсам сайта, созданное разрешение записывается в таблицу БД «AspNetUserRoles», Создание разрешение производится с использованием функционала приложения.

Для получения доступа к функционалу системы, пользователь обязан авторизоваться в системе, введя соответствующий логин и пароль. Пароли пользователей хранятся в базе данных предприятия в хешированном виде в таблице AspNetUsers.

# Вопросы информационной безопасности

## Анализ угроз

Под угрозами информационной безопасности понимается потенциальная возможность нарушения следующих основных характеристик:

* Конфиденциальности сведений, составляющих государственную, служебную или коммерческую тайну, а также персональных данных. Заключается в том, что информация становится известной тому, кто не располагает полномочиями доступа к ней;
* Целостности и достоверности информационных, программных и других ресурсов, а также фальсификация документов. Включает в себя любое умышленное изменение информации, хранящейся в вычислительной системе или передаваемой из одной системы в другую
* Доступности информации, в условиях случайных и преднамеренных угроз информация предоставляется в том виде и том месте, которые необходимы авторизованному (санкционированному) пользователю, и в то время, когда она ему необходима. Доступность определяется как наличием необходимой информации, так и готовностью системы к обслуживанию.

В отношении информационных систем принято считать, что информационная безопасность информационных систем обеспечена в том случае, если для любых информационных ресурсов в системе поддерживается определенный уровень: конфиденциальности, целостности, доступности.

Проведем анализ угроз и их оценку с точки зрения вероятности реализации и ущерба от реализации угрозы.

Оценка вероятности возникновения угрозы:

1. очень вероятна – 9-10 баллов,
2. вероятна – 5-8 баллов,
3. маловероятна – 3-5 баллов.
4. практически невероятна 1-2 балла.

Оценка степени ущерба возникновения угрозы:

1. полная потеря данных – 9-10 баллов,
2. частичная потеря данных – 3-8 балла,
3. возможная потеря данных – 1-2 балла.

В соответствии с ГОСТ Р 21275-2006 выделены следующие угрозы (таблица 13).

Таблица 13 – Анализ угроз

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Угрозы | Меры предотвращения | Вероятность возникновения | Ущерб | Коэффициент оценки угрозы |
| Разглашение защищаемой информации лицами, имеющими к ней право доступа. | Составление регламентов, должностных инструкций для пользователей системы. | 8 | 4 | 32 |
| Несанкционированный доступ к защищаемой информации | Контроль доступа в помещения | 7 | 5 | 35 |
| Блокирование доступа к защищаемой системе путем перегрузки тех. средств обработки информации | Особая организация работы систем с поступающей информацией | 4 | 3 | 12 |
| Искажение, уничтожение или блокирование информации с применением тех. средств | Экранирование элементов системы от внешних излучений, физических воздействий | 4 | 8 | 32 |
| Ошибки обслуживающего персонала | Увеличение степени автоматизированности процессов системы; обучение персонала. | 3 | 3 | 9 |
| Явления техногенного характера | Установка систем пожаротушения | 3 | 8 | 24 |
| Природные явления, стихийные бедствия | Заземление и установка громоотводов | 2 | 8 | 16 |

## Информационная безопасность на уровне предприятия.

### Контроль доступа в помещения.

Контроль доступа в помещения регламентируется общими правилами «КОМК». В частности, доступ к помещениям, в которых установлено серверное и коммутационное оборудование, осуществляется обученным и сертифицированным персоналом и ограничивается стандартными офисными средствами безопасности (ключ от помещения выдается под роспись).

### Обеспечение безопасности с помощью аппаратных средств.

* Все аппаратные средства имеют маркировку и занесены в журналы по учету оборудования предприятия;
* Все помещения КОМК оборудованы огнетушителями;
* Сотрудники колледжа ознакомлены с правилами безопасности использования аппаратных средств.

### Обеспечение безопасности с помощью программных средств.

* На рабочие станции запрещено устанавливать нелицензионное ПО;
* Перед началом использования ПО сотрудники КОМК обязаны пройти курс обучения по работе с данными;
* Перед введением ПО в эксплуатацию, оно в обязательном порядке проходит тестирование на наличие уязвимостей и соответствие требованиям безопасности.

### Определение политики управления доступом пользователей.

* Для персонала определены полномочия и права доступа к информации в соответствии с занимаемой должностью;
* Каждому сотруднику выдан личный логин и пароль, с помощью которого происходит аутентификация личности. С периодичностью в два месяца пароль меняется для всех сотрудников колледжа.
* Данные о логинах и паролях хранятся централизованно. Передача логина или пароля другому лицу запрещена.
* Пользователи блокируют свою учетную запись, когда не работают на ней.
* При вводе пароля он не должен отображаться в явном виде.

### Требования к административным обязанностям.

В соответствии «Должностной инструкцией администратора», документом, регламентирующим обязанности администратора, на нем лежат следующие обязанности:

* 1. Устанавливает на серверы и рабочие станции операционные системы и необходимое для работы программное обеспечение.
  2. Осуществляет конфигурацию программного обеспечения на серверах и рабочих станциях.
  3. Поддерживает в работоспособном состоянии программное обеспечение серверов и рабочих станций.
  4. Регистрирует пользователей локальной сети и почтового сервера, назначает идентификаторы и пароли.
  5. Осуществляет техническую и программную поддержку пользователей, консультирует пользователей по вопросам работы локальной сети и программ, составляет инструкции по работе с программным обеспечением и доводит их до сведения пользователей.
  6. Устанавливает права доступа и контролирует использование сетевых ресурсов.
  7. Обеспечивает своевременное копирование, архивирование и резервирование данных.
  8. Принимает меры по восстановлению работоспособности локальной сети при сбоях или выходе из строя сетевого оборудования.
  9. Выявляет ошибки пользователей и программного обеспечения и принимает меры по их исправлению.
  10. Проводит мониторинг сети, разрабатывает предложения по развитию инфраструктуры сети.
  11. Обеспечивает сетевую безопасность (защиту от несанкционированного доступа к информации, просмотра или изменения системных файлов и данных), безопасность межсетевого взаимодействия.
  12. Осуществляет антивирусную защиту локальной вычислительной сети, серверов и рабочих станций.
  13. Готовит предложения по модернизации и приобретению сетевого оборудования.
  14. Осуществляет контроль за монтажом оборудования локальной сети специалистами сторонних организаций.
  15. Сообщает своему непосредственному руководителю о случаях нарушения правил пользования локальной вычислительной сетью и принятых мерах.

Системный администратор несет ответственность за:

* 1. Нарушение функционирования локальной вычислительной сети, серверов и персональных компьютеров вследствие ненадлежащего исполнения своих должностных обязанностей.
  2. Несвоевременную регистрацию пользователей локальной вычислительной сети и почтового сервера.
  3. Несвоевременное уведомление руководства о случаях нарушения правил пользования локальной вычислительной сетью.
  4. Системный администратор привлекается к ответственности:
     1. За ненадлежащее исполнение или неисполнение своих должностных обязанностей, предусмотренных настоящей должностной инструкцией - в пределах, установленных действующим трудовым законодательством Российской Федерации.
     2. За правонарушения, совершенные в процессе своей деятельности - в пределах, установленных действующим административным, уголовным и гражданским законодательством Российской Федерации.
     3. За причинение материального ущерба компании - в пределах, установленных действующим законодательством Российской Федерации.

### Антивирусная защита информации.

* На каждой рабочей станции установлено антивирусное ПО;
* Внешние носители сканируются на наличие вирусного ПО;
* Отключение антивирусного ПО запрещено;
* Раз в неделю проводится сканирование рабочих станций на наличие вирусного ПО.

### Информационная безопасность на уровне АИС.

Так как рабочие станции на предприятии уже защищены от выделенных угроз, разрабатываемая информационная система не нуждается в дополнительной защите.

В разрабатываемой информационной системы будет использоваться механизм идентификации пользователей по логину и паролю, они хранятся в базе данных информационной системы в отдельной таблице БД.

Использование логина и пароля необходимо, так как в разрабатываемой системе будет несколько групп пользователей. В других мерах защиты информации система не нуждается, так как на самом предприятии предусмотрена парольная система аутентификации пользователей за рабочим местом.

### Защита персональных данных.

Защита персональных данных в Кемеровском областном медицинском колледже регламентируется «Положением о защите персональных данных». В соответствии с действующим положением действуют следующие правила:

1. **Получение персональных данных**
   1. При устройстве на работу работник обязан ознакомиться с действующим «Положением о защите персональных данных» и дать согласие на обработку своих персональных данных в письменном виде.
   2. Работник обязан предоставить колледжу достоверные паспортные данные. При смене паспортных данных, работник должен письменно уведомить об этом колледж в срок, не превышающий 14 дней. Колледж имеет право запросить у работника дополнительные сведения и документы, подтверждающие его личность.
2. **Обработка персональных данных**
   1. Допуск к персональным данным работника разрешен должностным лицам, которым персональные данные необходимы для выполнения конкретных трудовых функций, в частности: начальник отдела кадров и делопроизводства, специалисты по кадрам, инспектор по кадрам, главный бухгалтер, заместитель главного бухгалтера, начальник отдела организации и оплаты труда, бухгалтера, начальник отдела сопровождения системной инфраструктуры, сотрудники тех.поддержки и т.д.
   2. При обработке персональных данных, не связанных с исполнением трудового договора, колледж обязан получить согласие работника на обработку его персональных данных в письменном виде.
   3. Колледж обязан обеспечивать защиту персональных данных от несанкционированного доступа и копирования.
3. **Хранение персональных данных**
   1. Отдел кадров и делопроизводства организует хранение и использование персональных данных работников в соответствии с федеральным законом "О персональных данных" от 27.07.2006 N 152-ФЗ.
   2. Хранение персональных данных работников осуществляется на электронных носителях, а также в бумажном варианте.
   3. Доступ к программному обеспечению, а также к персональной информации, хранящейся на электронных носителях, осуществляется при введении личного идентификатора и пароля пользователя.
   4. Документы персонального характера хранятся в сейфе отдела кадров и делопроизводства.
4. **Уничтожение персональных данных**
   1. Документы, содержащие персональные данные, подлежат хранению и уничтожению в порядке, предусмотренном федеральным законом "О персональных данных" от 27.07.2006 N 152-ФЗ.
   2. Персональные данные работников подлежат уничтожению по достижении целей обработки или в случае утраты необходимости в достижении таких целей.

# Экономическое обоснование разработки ИС

## Расчет затрат на разработку системы

Проектные затраты на разработку системы определяются по формуле 9.1.

 (9.1)

где  – затраты на приобретение оборудования, руб.;

 – заработная плата исполнителям и разработчикам проекта, руб.;

– отчисления на страховые взносы, руб.;

 – затраты на электроэнергию, руб.;

 – накладные расходы, руб.;

 – затраты на амортизацию, руб.;

 – прочие затраты, руб.

1. Затраты на приобретение оборудования:

Т.к. требуемое для развертки приложения оборудование уже есть в наличии, то:

*Зоб = 0*

1. Затраты на заработную плату исполнителям и разработчикам проекта:

Разрабатывается программа проведения проектных работ, определяются ориентировочные трудозатраты этапов работ по исполнителям и стоимость этих затрат (человек-часов) по формуле 9.2

 (9.2)

где  – заработная плата *j*-го исполнителя;

 – ориентировочные трудозатраты *i*-го этапа *j*-го исполнителя в часах;

 – стоимость человека-часа *j*-го исполнителя.

Общие затраты на заработную плату определим по формуле 9.3.

, (9.3)

где *j* = 1…*m* – количество исполнителей.

*Сч\ч = Ом / (Крд \* Драб)*

где *Сч\ч* – стоимость человека-часа, руб.,

*Ом* – месячный оклад, руб.,

*Крд* – количество рабочих дней в месяце (равный двадцати одному),

*Драб* – 8-ми часовой рабочий день.

Средняя заработная плата программиста, занимающегося разработкой приложений составляет 30 000 руб. в месяц. Вычисляем стоимость человеко-часа:

*Сч\ч* = 30000/(21\*8) = 178,57 руб.

 (9.4)

где *Тор* – ориентировочные трудозатраты;

*Дi* – дни, затраченные на выполнение i-ого этапа работы;

*Драб* – 8-ми часовой рабочий день.

Работа ответственного за проект включает в себя следующие этапы, представленные в таблице 14:

Таблица 14 - Расчет ориентировочных трудозатрат

|  |  |  |  |
| --- | --- | --- | --- |
| № п/п | Этап работы | Затраченные дни | Ориентировочные трудозатраты чел-час, Top |
| 1 | Постановка задачи | 1 | 8 |
| 2 | Определение хода работы | 4 | 32 |
| 3 | Анализ поставленной задачи |
| 4 | Изучение объекта автоматизации |
| 5 | Составление плана работы |
| 6 | Обзор литературы, детальное изучение объекта | 2 | 16 |
| 7 | Написание общей части дипломного проекта и разработка информационной системы автоматизации | 30 | 240 |
| 8 | Расчет экономиче­ской эффективно­сти | 3 | 24 |
| 9 | Оформление проекта | 7 | 56 |
| итого: | | 47 | 376 |

Произведем расчет заработной платы по формуле 9.2:

Зз/п = *Сч\ч* \**Top=*178,57\*376 = 67142,32 руб.

Таким образом, общие затраты на заработную плату составляют 67142,32 рублей.

1. Отчисления в страховые взносы:

Страховы взносы составляют 30 % от заработной платы.

Зсв = Зз/п \* 0,3 = 95713,52 \* 0,3 = 20142,7 руб.

1. Затраты на электроэнергию:

, (9.5)

где – потребляемая мощность *i*-го токоприемника, кВт/ч;

 – фактическое время работы *i*-го токоприемника, час;

 – цена на электроэнергию за 1 кВт/час, руб.;

 – коэффициент использования по времени *i*-го токоприемника, %.

 = 0,25 кВт/ч;

= 376 часа;

 = 3,04 руб;

 = 0,93.

Подставляя значения в формулу 9.5, получим

Зэн = 0,25 \* 376 \* 3,04 \* 0,93 = 265,76 рублей

5. Накладные расходы:

Накладные расходы принимаются в размере 10 % от заработной платы исполнителей

Зн.р = 67142,32 \* 0,1 =6714,23 руб.

6. Расходы на амортизацию:

Расходы на амортизацию составляют 12,5 % от стоимости оборудования:

Система, развернутая на предприятии использует для функционирования:

* Сервер HP ProLiant ML150 Gen9 776274-421 стоимостью 71 944 рублей;
* Рабочая станция сотрудника отдела тех. обслуживания:
  + ПК DEXP Aquilon O100, стоимостью 9 890 рублей;
  + Монитор ViewSonic VA1903a [VS16216], стоимостью 3 990 рублей;
  + Мышка и клавиатура общей стоимостью 500 рублей.

Цоб = 71944 + 9890 + 3990 + 500 = 86324 рублей.

Зам = Цоб \* 0,125 = 86324 \* 0,125 = 10790,5 руб.

7. Прочие затраты:

Прочие затраты составляют 3 % от суммы всех предшествующих затрат

. (9.6)

 = 0,03 \* (0 + 67142,32 + 20142,7 + 265,76 + 6714,23 + 10790,5) = 3151,67руб.

8. Затраты на этапе проектирования.

Смета проектных затрат приведена в таблице 14:

Таблица 15 – Таблица затрат

|  |  |  |
| --- | --- | --- |
| пп/п | Наименование затрат | Сумма, руб. |
| 1 | Расходы на оборудование (Зоб ) | 0 |
| 2 | Расходы на заработную плату (Зз/п ) | 67142,32 |
| 3 | Отчисления на страховые взносы (ЗСВ) | 20142,7 |
| 4 | Затраты на электроэнергию (Зэн ) | 378,84 |
| 5 | Накладные расходы (Зн.р) | 6714,23 |
| 6 | Затраты на амортизацию (Зам) | 10790,5 |
| 7 | Прочие расходы (Зпр) | 3151,67 |
|  | Итого | 108320,26 |

Итого затраты на проектирование составят 108320,26 рублей.

## Расчёт эксплуатационных затрат

Затраты на внедрение системы вычисляются по формуле 9.7.

, (9.7)

где  – эксплуатационные затраты, руб.;

 – затраты на оборудование, руб.;

 – затраты на ремонт и содержание оборудования, руб.;

 – затраты на заработную плату обслуживающего персонала, руб.;

 – отчисления на страховые взносы, руб.;

 – расходы на энергию для эксплуатации оборудования, руб.;

 – затраты на амортизацию, руб.;

 – накладные расходы, руб.;

 – прочие расходы, руб.

1. Затраты на оборудование:

, (9.8)

где – цена оборудования, руб.;

 – транспортные расходы (10 % от), руб.;

 – расходы, связанные с монтажом оборудования (8 % от), руб.

 = 0 руб;

 = 0,1\*0= 0 руб;

= 0,08\*0= 0 руб.

Подставляя значения в формулу 9.11, получим:

 = 0 руб.

2. Затраты на ремонт:

Затраты на ремонт и содержание оборудования принимаются в размере 2,5 % от стоимости основных средств

. (9.9)

 = 86324 \* 0,025 = 2158,1 руб.

3. Затраты на заработную плату обслуживающего персонала:

Затраты на заработную плату будут состоять из заработной платы специалиста по ремонту специально предназначенного для обслуживания ЭВМ и оборудования. ЭВМ работает весь рабочий день. Количество выходов в месяц специалиста по ремонту составит: 30 – 9 = 21 день.

Количество дней в году 365, из них 117 – выходные и праздничные дни, 28 – отпуск. Таким образом действительный фонд рабочего времени составляет 220 дней.

Среднюю заработную плату сервисного инженера, обслуживающего оборудование, рассчитаем по формуле 9.10:

 (9.10)

где *D* – годовой должностной оклад работника, руб.;

*К* – коэффициент, учитывающий премиальный (Кпр = 1,4) и районный (Кр = 1,3) коэффициенты.

Годовые затраты на заработную плату составят:

= 23 000 \* 12 \* 1,4 \* 1,3 = 502320 руб.

1. Отчисления в :

Страховы взносы составляют 30 % от заработной платы.

ЗСВ = Зз/п \* 0,3 = 502320 \* 0,3 = 150696 руб. (9.11)

1. Затраты на электроэнергию:

Расходы на электроэнергию рассчитывают по стоимости потребляемой токоприемником электроэнергии

, (9.12)

где *Р* – средняя потребляемая мощность, кВт/час.;

*Т* – время работы токоприемника, час.;

 – коэффициент использования по мощности (0,92–0,95);

 – коэффициент использования по времени (0,92–0,95);

 – цена за электроэнергию за 1 кВт∙час., руб.

*Р =* 0,25 кВт/час;

*Т* = 220\*8 = 1760 часов;

 ≈ 0,93;

 ≈ 0,94;

 = 3.04 руб.

= 0,25\*1760\*0,93\*0,93\*3,04 = 1156,89 руб.

1. Накладные расходы:

Накладные расходы принимаются в размере 10 % от заработной платы обслуживающего персонала.

= 502320 \* 0,1 = 50232 руб.

1. Расходы на амортизацию:

Расходы на амортизацию составляют 12,5 % от стоимости оборудования.

 = Цоб \* 0,125 = 86324 \* 0,125 = 10790,5 руб.

1. Прочие затраты:

Прочие затраты составляют 3 % от суммы всех предшествующих затрат

и рассчитываются по формуле 9.13.

 = 0,03( + + +  +  +  + ) (9.13)

= 0,03 \* (0 + 2158,1 + 502320 + 150696 + 1156,89 + 10790,5 + 50232) = 21520,6 руб.

1. Затраты на этапе эксплуатации.

Смета проектных затрат приведена в таблице 16.

Таблица 16 – Смета затрат

|  |  |  |
| --- | --- | --- |
| п/п | Наименование затрат | Сумма, руб. |
| 1 | Расходы на оборудование (Зоб) | 0 |
| 2 | Расходы на заработную плату (Зз / п) | 502320 |
| 3 | Затраты на ремонт (Зрем) | 2158,1 |
| 4 | Отчисления на единый социальный налог (ЗСВ) | 150696 |
| 5 | Расход на электроэнергию (Зэн) | 1156,89 |
| 6 | Накладные расходы (Зн.р) | 50232 |
| 7 | Расходы на амортизацию (Зам) | 10790,5 |
| 8 | Прочие расходы (Зпр) | 21520,6 |
|  | Итого | 738874,09 |

Итого затраты на эксплуатацию составят 738874,09 рублей.

## Экономическая эффективность

Экономическая эффективность внедрения системы осуществляется за счет повышения эффективности разработанной ИС: повышения производительности и снижения эксплуатационных затрат по сравнению с существующей системой.

При одинаковой производительности систем ориентировочно годовая экономия составит



где Зэкс.анал.сист. – эксплуатационные затраты аналогичной системы в случае ее приобретения (или существующей); Зэкс.проект.сист. – эксплуатационные затраты проектируемой системы.

Возьмем за аналогичную систему R-Keeper. Стандартная стоимость системы за 6 лицензий за год 120 000 руб. При условии, что затраты на эксплуатацию систем, а именно расходы на заработную плату, единый социальный налог и прочее равны для обеих систем получаем, что годовая экономия равняется стоимости годовой лицензии:

Эгод = Зэкс.анал.сист. + Згод.лиценз. - Зэкс.проект.сист.= 738874,09 + 120 000 - 738874,09 = 120 000 руб.

## Срок окупаемости

Срок окупаемости – период времени, необходимый для того, чтобы доходы покрыли затраты.

Срок окупаемости информационной системы определяется как период времени, в течении которого экономический эффект от эксплуатации системы сравнивается с капитальными затратами на данную систему.

Срок окупаемости определяется по формуле 9.19.

*Т = Зобщ / Эгод* (9.19)

где *Т* – срок окупаемости, лет;

*Зобщ* - суммарные затраты на разработку и внедрение системы, руб.

*Эгод*– годовая экономия, руб

Т = 108320,26*/*120 000 = 0,9 года

Проанализировав экономические показатели, можно сделать вывод о целесообразности разработки информационной системы автоматизации процесса учёта деятельности на комбинате питания и внедрения ее в эксплуатацию, поскольку максимальный срок окупаемости составляет 0,9 года, что не превышаетнормативный срок окупаемости информационных технологий, равный 3 года.

## Технико-экономические показатели проекта

Технико-экономические показатели проекта представлены в таблице 16.

Таблица 16 – Технико–экономические показатели проекта

|  |  |  |
| --- | --- | --- |
| Наименование показателя | Единица измерения | Значение показателя |
| Проектные затраты | | |
| 1.1. Затраты на оборудование | руб | 0 |
| 1.2. Заработная плата исполнителям и разработчикам проекта | руб | 67 142,32 |
| 1.3. Отчисления в страховые взносы | руб | 20 142,7 |
| 1.4. Затраты на электроэнергию | руб | 378,84 |
| 1.5. Накладные расходы | руб | 6 714,23 |
| 1.6. Затраты на амортизацию | руб | 10 790,5 |
| 1.7. Прочие затраты | руб | 3 151,67 |
| Эксплуатационные затраты | | |
| 1.8. Затраты на оборудование | руб | 0 |
| 1.9. Затраты на ремонт и содержание оборудования | руб | 2 158,1 |
| 1.10. Затраты на заработную плату обслуживающего персонала | руб | 502 320 |
| 1.11. Отчисления в страховые взносы | руб | 150696 |
| 1.12. Расходы на энергию для эксплуатации оборудования | руб | 1156,89 |
| 1.13. Затраты на амортизацию | руб | 10790,5 |
| 1.14. Накладные расходы | руб | 50232 |
| 1.15. Прочие расходы | руб | 21520,6 |
| 2. Всего затрат | руб | 847 194,35 |
| 3. Экономическая эффективность | руб | 120 000 |
| 4. Срок окупаемости | года | 0,9 |

Таким образом стоимость экономических затрат составила 847194,35 рублей со сроком окупаемости в 0.9 года.

Список используемой литературы

1. Ванеев О.Н. Селезнев В.В. Методические указания по выполнению курсового проекта по дисциплине "Управление данными" для студентов 3 курса специальности 071900 (230201) "Информационные системы и технологии". Кемерово, КузГТУ 2014.
2. Ванеев О.Н. Программа преддипломной практики для студентов специальности 230201 "Информационные системы и технологии". Кемерово, КузГТУ 2015.
3. Ванеев О.Н., Полетаев В.А. Методические указания по дипломному проектированию для студентов специальности 230201 "Информационные системы и технологии". Кемерово, КузГТУ 2015.
4. Ванеев О.Н., Методические указания по выполнению курсового проекта по дисциплине "Проектирование информационных систем" для бакалавров направления 09.03.02 Информационные системы и технологии / О. Н. Ванеев –Кемерово, КузГТУ 2014.
5. Шилдт Г. “C# 4.0 Полный справочник”. Пер. с англ. –М.: ООО “И.Д.Вильямс”, 2011-1056с.
6. Microsoft Developer Network //Microsoft[Электронный ресурс].-URL: <http://msdn.microsoft.com>
7. Хабрахабр [Электронный ресурс].- URL: <http://habrahabr.ru>
8. METANIT.COM [Электронный ресурс].- URL: <https://metanit.com>
9. ProfessorWeb [Электронный ресурс].- URL: https://professorweb.ru

Заключение

В процессе выполнения выпускной квалификационной работы решены все поставленные задачи.

Для выявления места разрабатываемой системы в общей деятельности предприятия, уточнения цели разрабатываемой системы был проведен анализ деятельности предприятия, его целей и функций.

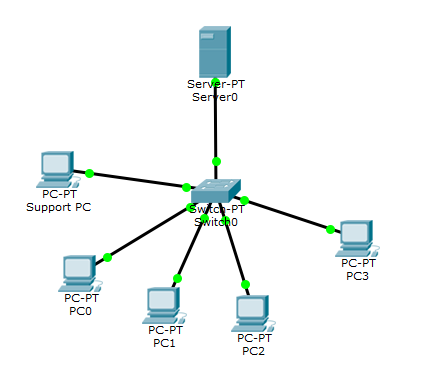
Решены вопросы, связанные с:

* Разработкой технического задания;
* Анализом и проектированием;
* Разработкой системы централизованного хранения и обработкой данных;
* Специальной частью;
* Технологии разработки и программная реализация;
* Аппаратная и административная интеграция ИС;
* Общие вопросы администрирования;
* Вопросы информационной безопасности;
* Экономическое обоснование разработки ИС.

Результатом работы является информационная система автоматизации процесса учёта деятельности на комбинате питания ГБПОУ «Кемеровский областной медицинский колледж» г. Кемерово. Система позволяет сократить временные затраты с учетом средств вычислительной техники, а также минимизировать шанс человеческой ошибки.

Разработанная ИС решает задачи конкретного предприятия. В чем и заключается ее основное достоинство по сравнению с альтернативными системами.

ПРИЛОЖЕНИЕ 1. Схема ЛВС



ПРИЛОЖЕНИЕ 2. Организационная структура колледжа

ПРИЛОЖЕНИЕ 3. Схема управления организационными структурами КОМК

