

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

федеральное государственное бюджетное образовательное учреждение  
высшего образования  
«Тольяттинский государственный университет»

Институт математики, физики и информационных технологий  
(институт)

Прикладная математика и информатика  
(кафедра)

09.03.03 Прикладная информатика  
(код и наименование направления подготовки, специальности)

Бизнес-информатика  
(наименование профиля, специализации)

## БАКАЛАВРСКАЯ РАБОТА

на тему Разработка информационной системы технической поддержки на  
основе методологии ITIL

Студент	<u>О.А. Иванов</u> (И.О. Фамилия)	_____	(личная подпись)
Руководитель	<u>Е.В. Панюкова</u> (И.О. Фамилия)	_____	(личная подпись)

**Допустить к защите**

Заведующий кафедрой к.т.н., доцент, А.В. Очеповский  
(ученая степень, звание, И.О. Фамилия) \_\_\_\_\_  
(личная подпись)

« \_\_\_\_\_ » \_\_\_\_\_ 20\_\_ г.

Тольятти 2019



**Росдистант**  
ВЫСШЕЕ ОБРАЗОВАНИЕ ДИСТАНЦИОННО

## АННОТАЦИЯ

Тема выпускной квалификационной работы «Разработка информационной системы технической поддержки на основе методологии ITIL».

Целью данной работы является разработка информационной системы технической поддержки на основе методологии ITIL.

Объектом данной работы является служба технической поддержки (Service desk) автомобильной сервисной компании.

Предметом данной работы является автоматизация службы технической поддержки (Service desk) автомобильной сервисной компании.

В первой главе дана технико-экономическая характеристика предметной области, разработана модель бизнес-процесса управление инцидентами «КАК ДОЛЖНО БЫТЬ» в соответствии с методологией ITIL, определена задача по разработке и внедрению системы ServiceDesk.

Во второй главе было описано логическое проектирование разрабатываемой информационной системы, произведен выбор технологии логического моделирования, построены диаграммы вариантов использования, спроектирована логическая модель данных и выработаны требования к аппаратно-программному обеспечению разрабатываемой системы, приведены примеры ее интерфейсов.

В третьей главе выпускной квалификационной работы дана оценка и обоснование экономической эффективности разработки информационной системы технической поддержки на основе методологии ITIL.

В работе 73 страницы, 25 рисунков, 18 таблиц.

## ОГЛАВЛЕНИЕ

ВВЕДЕНИЕ.....	5
Глава 1 Анализ бизнес-процессов в структуре управления автомобильной сервисной компании .....	7
1.1 Технико-экономическая характеристика предметной области.....	7
1.1.1 Характеристика предприятия .....	7
1.1.2 Сущность задачи автоматизации.....	9
1.2 Разработка и анализ модели бизнес-процесса управление инцидентами «КАК ЕСТЬ».....	11
1.3 Разработка и анализ модели бизнес-процесса управление инцидентами «КАК ДОЛЖНО БЫТЬ» в соответствии с ITIL .....	14
1.4 Постановка задачи по разработке и внедрению ServiceDesk системы.....	16
1.4.1 Цель и назначение автоматизированного варианта решения задачи .....	16
1.4.2 Формализация моделирования процессов и подпроцессов, реализуемых системой.....	19
1.5 Анализ существующих разработок и обоснование выбора технологии проектирования .....	22
Выводы по главе 1 .....	24
Глава 2 Разработка и реализация проектных решений по проектированию, разработке и внедрению service desk в службу технической автомобильной сервисной компании .....	25
2.1 Логическое моделирование предметной области.....	25
2.1.1 Логическая модель и ее описание .....	25
2.1.2 Характеристика базы данных .....	30
2.1.3 Характеристика результатной информации.....	33
2.2 Физическое моделирование АИС.....	36

2.2.1 Выбор архитектуры АИС .....	36
2.2.2 Функциональная схема проекта .....	37
2.2.3 Разработка физической модели данных ServiceDesk .....	39
2.2.4 Структурная схема проекта.....	42
2.2.5 Описание программных модулей.....	44
2.2.6 Схема взаимосвязи программных модулей и информационных файлов...	47
2.3 Контрольный пример реализации проекта и его описание .....	52
Выводы по главе 2.....	58
Глава 3 Оценка и обоснование экономической эффективности проекта .....	60
3.1 Расчет показателей экономической эффективности проекта.....	60
Выводы по главе 3.....	71
ЗАКЛЮЧЕНИЕ .....	73
СПИСОК ИСПОЛЬЗУЕМОЙ ЛИТЕРАТУРЫ .....	74

## ВВЕДЕНИЕ

*Актуальность.* Одним из основных условий развития бизнеса в современных условиях является оптимизация бизнес-процессов. Для эффективной работы любого предприятия необходимо внедрение разного рода информационных систем, направленных на автоматизацию деятельности как одного отдела (CRP), так и всего предприятия в целом (ERP).

Использование ИТ технологий всегда актуально в таких сферах деятельности любого предприятия как ведение бухгалтерии и управление финансами, управление клиентской базой и базой поставщиков, управление и автоматизация технологических процессов. Еще одним, очень важным направлением внедрения ИТ согласно ITIL является организации и поддержка обслуживания средств ИТ на предприятии.

ITIL – библиотека инфраструктуры информационных технологий, описывает лучшую мировую практику организации предприятия или подразделения, предоставляющая услуги в сфере информационных технологий. Методологическое изложение ITIL позволяет обеспечить эффективное функционирование ИТ-служб, удовлетворить потребности бизнес-пользователей, обеспечить стабильное и предсказуемое развитие информационной системы предприятия.

Способность современного предприятия быть успешным на современном рынке во многом зависит от надежности используемых ею информационных технологий.

В нашем случае речь идет об ИТ службе автомобильной сервисной компании среднего размера, управление которой принципиально не отличается от аналогичной задачи любого другого предприятия, хотя имеет свои особенности связанные с направлением деятельности фирмы и формой собственности.

**Цель:** Разработка информационной системы технической поддержки на основе методологии ITIL.

**Предметная область** - системы автоматизации предприятия.

**Объект исследования** – служба технической поддержки (Service desk) автомобильной сервисной компании.

**Предмет исследования** - автоматизация службы технической поддержки (Service desk) автомобильной сервисной компании.

**Задачи исследования:**

1. Проанализировать возможности автоматизации службы технической поддержки автомобильной сервисной компании.
2. Определить основные функции системы и экономическую целесообразность ее внедрения.
3. Построить модели ИТ службы автомобильной сервисной компании.
4. Выполнить проектирование автоматизированной службы технической поддержки.
5. Выполнить разработку автоматизированной службы технической поддержки автомобильной сервисной компании.

**Методы исследования:**

- Методы научного познания, которые использовались при работе над теоретически-аналитическим блоком работы: анализ, синтез, индукция, выделение частного, обобщение, наблюдение, описание, аналогия.
- Общенаучные методы познания, которые выполнялись при работе над практической частью работы: Абстрагирование, моделирование, анализ, описание, эксперимент, формализация, идеализация.

# Глава 1 Анализ бизнес-процессов в структуре управления автомобильной сервисной компании

## 1.1 Технико-экономическая характеристика предметной области

### 1.1.1 Характеристика предприятия

ООО «ЛенАвтоТехника» крупная автомобильная сервисная компания, предоставляющая полный комплекс услуг оказывая полный комплекс услуг, среди которых эвакуация, экстренная помощь на дороге, выездной шиномонтаж и другие работы.

Основной офис компании находится в г. Санкт-Петербурге. В компании существует круглосуточный многоканальный колл-центр и диспетчерская служба. Большое внимание в компании уделяется качеству предоставляемых услуг, пропагандируется профессионализм и индивидуальный подход к каждому клиенту.

ООО «ЛенАвтотехника» входит в группу компаний «ЛАТ», которая работает уже более 25 лет. Опыт и знания, накопленные специалистами компании, позволяют оказывать различные услуг высокого уровня. Основной сайт компании – lat.spb.ru. Схема организационной структуры «ООО ЛенАвтоТехника» представлена на рисунке 1.1.

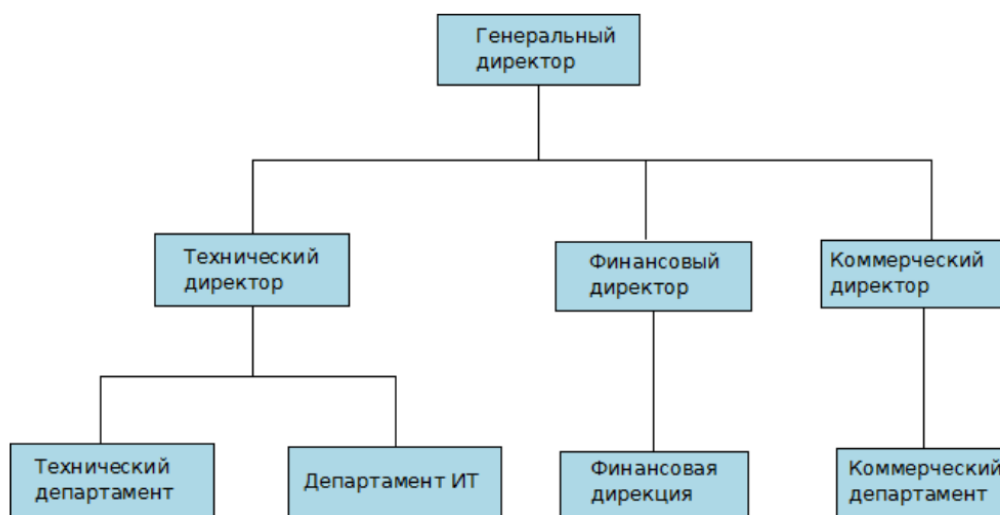


Рисунок 1.1 - Организационная структура предприятия

Охарактеризуем уровень автоматизации, существующий на предприятии. Рабочие места специалистов отделов объединены в локальную сеть и имеют доступ к общему локальному серверу.

Таблица 1.1 - Основные параметры локальной сети службы технической поддержки

<b>Характеристика</b>	<b>Значение</b>
Количество рабочих станций, всего:	140
Количество ПК, одновременно работающих в сети	95
Сетевые принтеры или МФУ	22 (принтер и МФУ)
Наличие и форма связи с удаленными объектами	Связь с пользователями сторонних органов местного самоуправления по удаленному подключению
Количество рабочих станций на удаленном объекте	Около 20
Характеристики компьютеров	Intel (TM) Core i5 Intel (TM) Core i5
Операционная система	Windows 7, 10
Используемые системы	Информационная система "КонсультантПлюс" Система электронного делопроизводства и документооборота "Кодекс: Документооборот"
Используемое ПО	Microsoft Office 2016 Kaspersky Endpoint Security Vip Net Client Крипто ПРО

Таблица 1.2 - Технические характеристики рабочей станции ИТ-специалиста

<b>Характеристика</b>	<b>Значение</b>
Периферия	Есть
Монитор	Philips 243V, 24', 1920x1080 (16:9), 6 мс, LED, 250 кд/м <sup>2</sup>
Описание	Офисный ПК
Процессор	Intel (TM) Core i5
Память	8 Gb
HDD	500 Gb 7200 об/мин 32 Мб
Оптический накопитель	DVD-ROM + CD-RW (COMBO) 52/32/52+16
Видеосистема	128Мб встроенная
FDD	нет



Технология работы подразделений автомобильной сервисной компании «ООО ЛенАвтоТехника» предполагает использование сетевых решений в рамках функционирования информационной системы. Перечень информационных систем, требующих использования сетевых ресурсов, приведен в таблице 1.3.

Таблица 1.3. Перечень информационных систем, требующих использования сетевых технологий

<b>Наименование программного продукта</b>	<b>Технологическое решение</b>	<b>Расположение</b>
1С 8.3: Бухгалтерский Учет	Тонкий клиент	Сервер 1С:Предприятие
1С 8.3: Зарплата и Управление Персоналом	Тонкий клиент	Сервер 1С:Предприятие
Учет вычислительной техники	Web-интерфейс	Web-сервер предприятия с сайтом предприятия
Учет криптовалют	Web-интерфейс	Web-сервер предприятия с сайтом предприятия
Складской учет	Тонкий клиент	Сервер 1С:Предприятие
Сдача отчетности в государственные органы	СБИС++	Сервер налоговой администрации
Работа с криптовалютами	Крипто-Про	Локальные рабочие станции пользователей
Система исполнения регламентов	Web-интерфейс	Web-сервер предприятия с сайтом предприятия
Единая система электронного документооборота и делопроизводства	Web-интерфейс	Сервер документооборота предприятия

Как видно из таблицы 1.3. существующая технология работы специалистов предполагает использование разнородных сетевых решений, обеспечивающих работу как с локальными базами данных, с информационными ресурсами, находящимися на удаленных серверах, а также с ресурсами Интернета.

### 1.1.2 Сущность задачи автоматизации

Из множества задач (регистрация и мониторинг состояния и количества оборудования в подразделениях управления предприятием «ООО

ЛенАвтоТехника», мониторинг сведений о размещении оборудования по подразделениям предприятия и о комплектующих частях и программном обеспечении, которые установлены в оборудовании; ведение учета о движении оборудования внутри предприятия, учет оборудования на рабочих местах специалистов, а также отслеживание и регистрация изменений в оборудовании и ПО после выполнения сервисных работ, учет поврежденного оборудования, формирование заявок на ремонт и обслуживание оборудования, контроль и мониторинг выполнения заявок, учет установленного программного обеспечения, составление отчетов о проведенных и запланированных работах), стоящих перед специалистами службы технической поддержки предприятия, была выбрана задача технической поддержки пользователей.

Именно бизнес-процесс «обработка заявки на ИТ обслуживание» является ключевым для службы технической поддержки предприятия, поскольку является основной частью процесса управления инцидентами согласно библиотеки ITIL. Благодаря этому процессу осуществляется обеспечение работы всех служб управления предприятия, автоматизация данного процесса ускорит процесс обслуживания.

Автоматизированная система службы технической поддержки должна обладать следующими функциями:

- авторизации;
- регистрации заявок;
- распределение заявок;
- выполнение заявок;
- контроль исполнения;
- формирование отчетности.

В автоматизации службы технической поддержки предприятия существует несколько классических решений Helpdesk и ServiceDesk.

Основные задачи Helpdesk системы в оказании помощи пользователям, в то время как ServiceDesk направлен на предоставление им комплекса ИТ услуг, чаще всего связанных с сопровождением различного вида заявок, реже

документов, то есть цель данной системы значительно шире и тесно связана с функцией обслуживания клиентов.

Основная функция Helpdesk – решение инцидентов, термин из библиотеки ITIL, а ServiceDesk, не только решает инциденты, но и управляет запросами на обслуживание (предоставление новых услуг) и запросами о предоставлении информации (находит средства для решения проблемы).

Другими словами, ServiceDesk шире, решает целый комплекс задач, поэтому термин применяется в ITIL значительно чаще, чем Helpdesk.

Предприятие предоставляет различные сервисные услуги автомобилистам, эта деятельность включает в себя множество бизнес-процессов. Часть из этих процессов сложно автоматизировать, но все управляющие бизнес-процессы, управление складской, внутренней и транспортной логистикой, бухгалтерию, документооборот и т.д. возможно и необходимо автоматизировать. Процесс автоматизации предполагает наличие рисков, связанных с отказами компьютерной техники. Для уменьшения рисков службой технической поддержки используется система ServiceDesk.

Данная система может использоваться более широко для автоматической обработки заявок различных служб и общения внутри предприятия.

## **1.2 Разработка и анализ модели бизнес-процесса управление инцидентами «КАК ЕСТЬ»**

На основе текущей ситуации в компании, построим и проанализируем модель бизнес-процесса управление инцидентами. Представим структуру и схемы бизнес-процесса в нотации IDEF0, соответствующие работе начальника департамента информационных технологий, старшего системного администратора, системного администратора и специалиста по телефонии. Ширина охвата описываемой системы определяется в пределах департамента ИТ ООО «ЛенАвтоТехника», а глубина детализации, определяющая степень подробности декомпозиции блоков и достигает 2-х уровней. Контекстная диаграмма деятельности департамента ИТ показана рисунке 1.2.

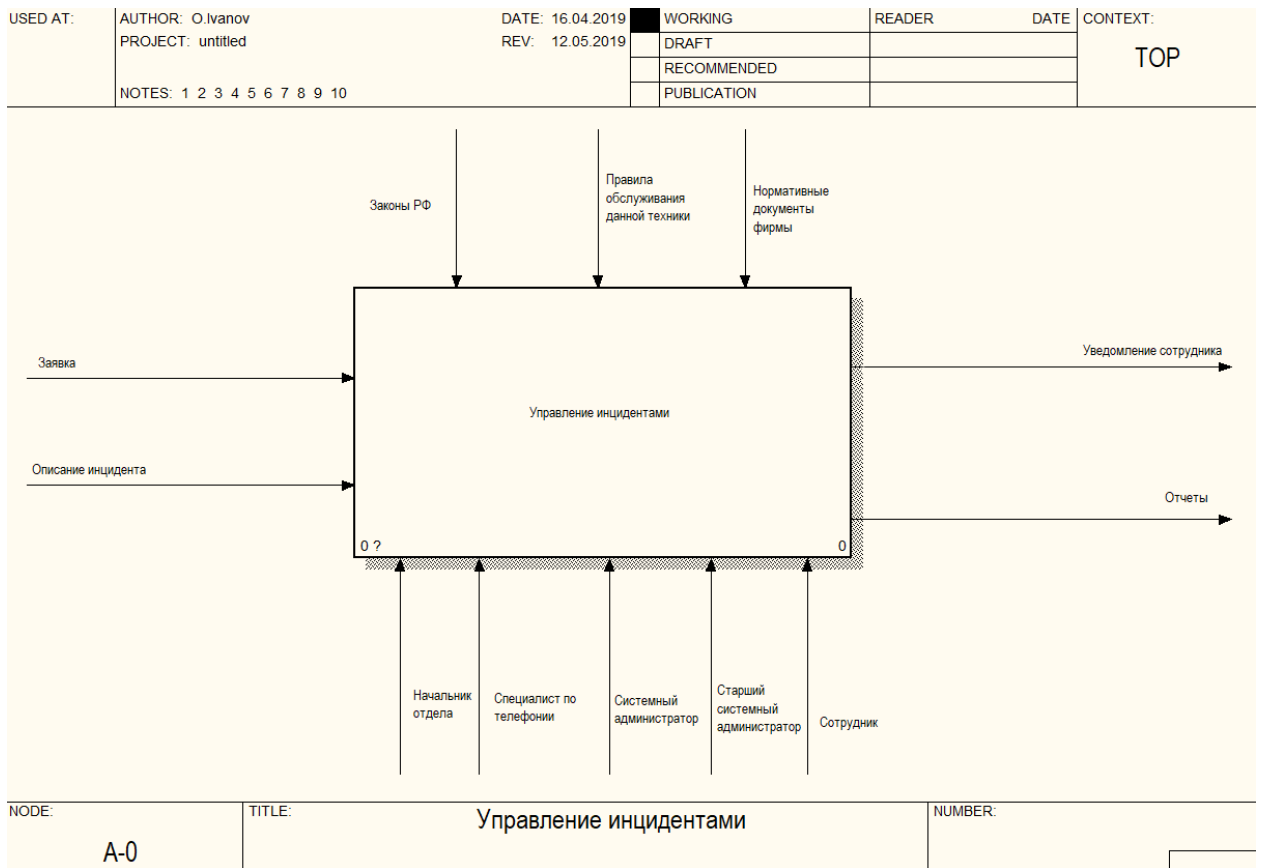


Рисунок 1.2 – Контекстная IDEF0-диаграмма бизнес-процесса управление инцидентами КАК ЕСТЬ (0-й уровень)

Входные информационные потоки, которые поступают в процессе деятельности департамента ИТ, которые можно считать данные по заявке, описание инцидента.

Управляющие потоки регламентируют выполнение бизнес-процессов, сюда относятся: законы РФ, правила обслуживания техники, нормативные документы.

Исполнителями всех рассматриваемых бизнес-процессов являются старший системный администратор, системный администратор, специалист по телефонии, руководитель департамента ИТ.

К результирующим данным относятся документы, которые позволяют анализировать выполнение заявок на ИТ обслуживание, уведомление пользователей об окончании работ.

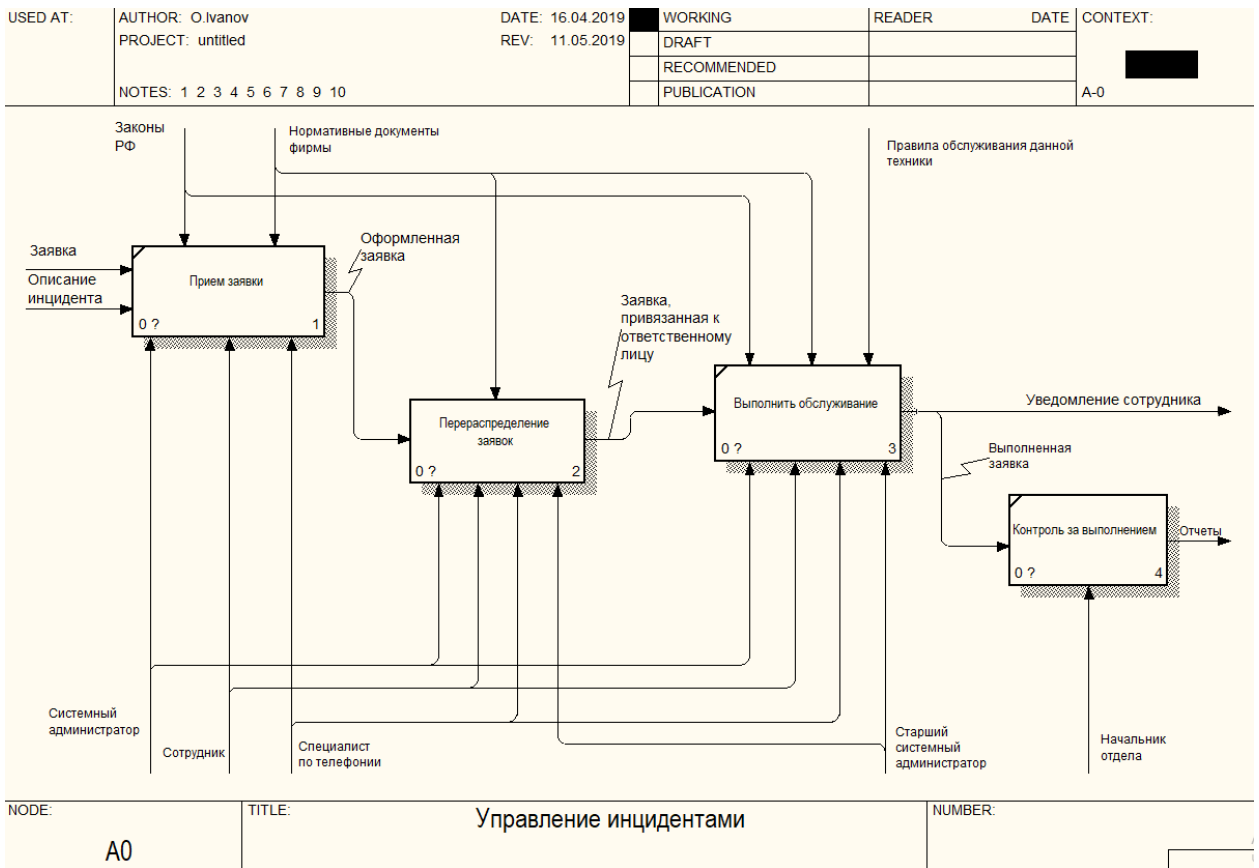


Рисунок 1.3 – Контекстная IDEF0-диаграмма бизнес-процесса управление инцидентами КАК ЕСТЬ (1-й уровень)

Обработку заявок на ИТ обслуживание выполняют системный администратор, старший системный администратор и специалист по телефонии. Заявки поступают по телефону и электронной почте, распределятся на усмотрение сотрудников или случайным образом.

Анализ модели КАК ЕСТЬ показал, что существующая система имеет следующие недостатки:

- нет единого центра приема заявок;
- заявка сразу не попадает на ответственного сотрудника;
- отсутствует регистрация заявок;
- отсутствует типизация заявок.

Такая ситуация не позволяет осуществлять оперативное выполнение заявок и эффективно контролировать их выполнение.

### 1.3 Разработка и анализ модели бизнес-процесса управление инцидентами «КАК ДОЛЖНО БЫТЬ» в соответствии с ITIL

Исходя из задач к ServiceDesk системе можно смоделировать ее с точки зрения обработки заявки на обслуживание ИТ. Контекстная IDEF0-диаграмма процесса обработки заявок на ИТ согласно ITIL показана на рисунке 1.4.

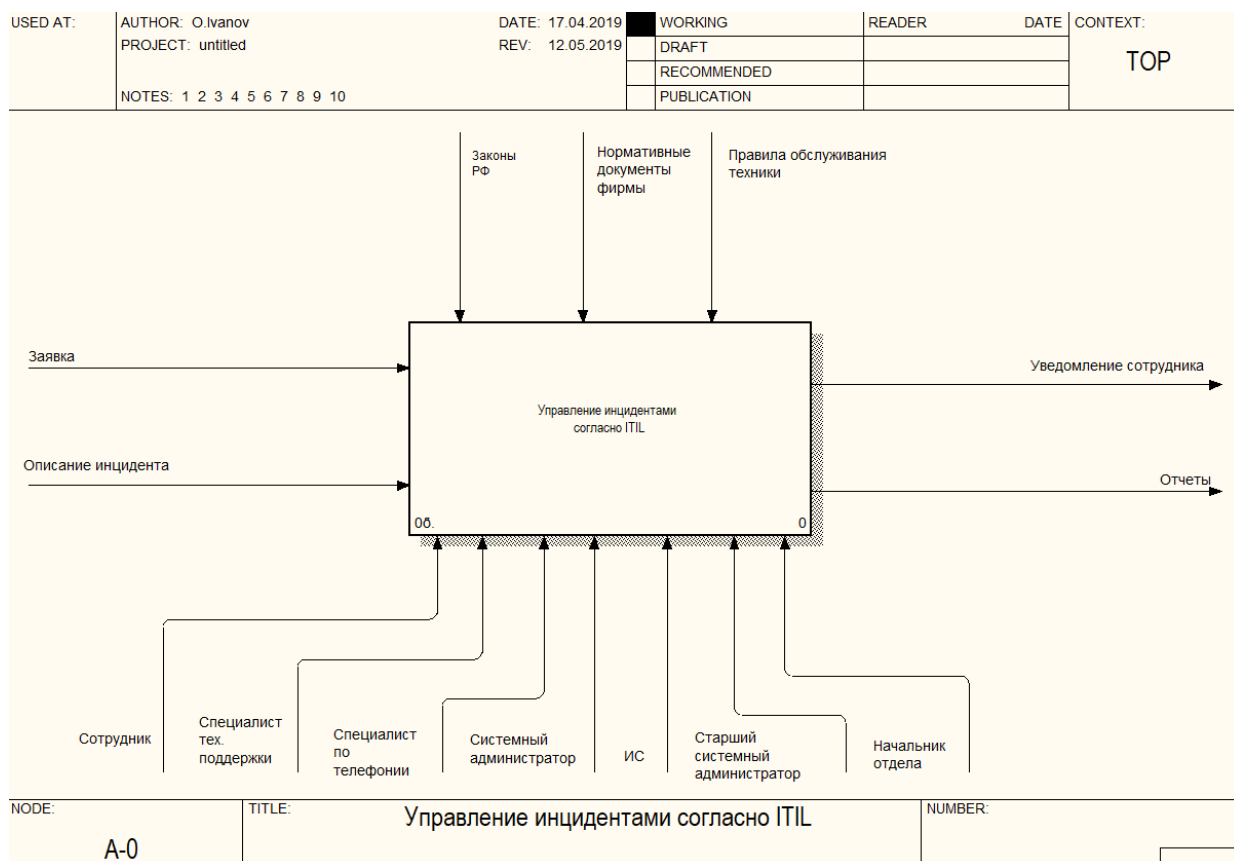


Рисунок 1.4 - Контекстная IDEF0-диаграмма бизнес-процесса управление инцидентами согласно ITIL (0-й уровень)

Декомпозиция бизнес-процесса управление инцидентами согласно ITIL представлена на рисунке 1.5.

Бизнес-процесс управление инцидентами согласно ITIL разбит на шесть блоков:

- прием заявки (сотрудник заполняет необходимые поля в системе по заявке и описывает инцидент, специалист технической поддержки принимает поступившую заявку);

- регистрация в системе (специалист тех. поддержки регистрирует поступившую заявку в ServiceDesk, устанавливая статус заявки в «в работе»);
- решение типовых проблем (специалист тех. поддержки выполняет самостоятельное решение типовых проблем, если это возможно);
- назначение ответственного специалиста (специалист тех. поддержки определяет тип инцидента и назначает заявку на соответствующего специалиста);
- обработка заявки (специалист осуществляет обработку поступившей заявки, после успешного решения меняет статус заявки на «выполнено», уведомляет сотрудника о завершении работ);
- контроль за выполнением (начальник отдела осуществляет контроль за выполнением работ через систему ServiceDesk, может оперативно видеть статусы и типы заявок, строить отчеты по заявкам).

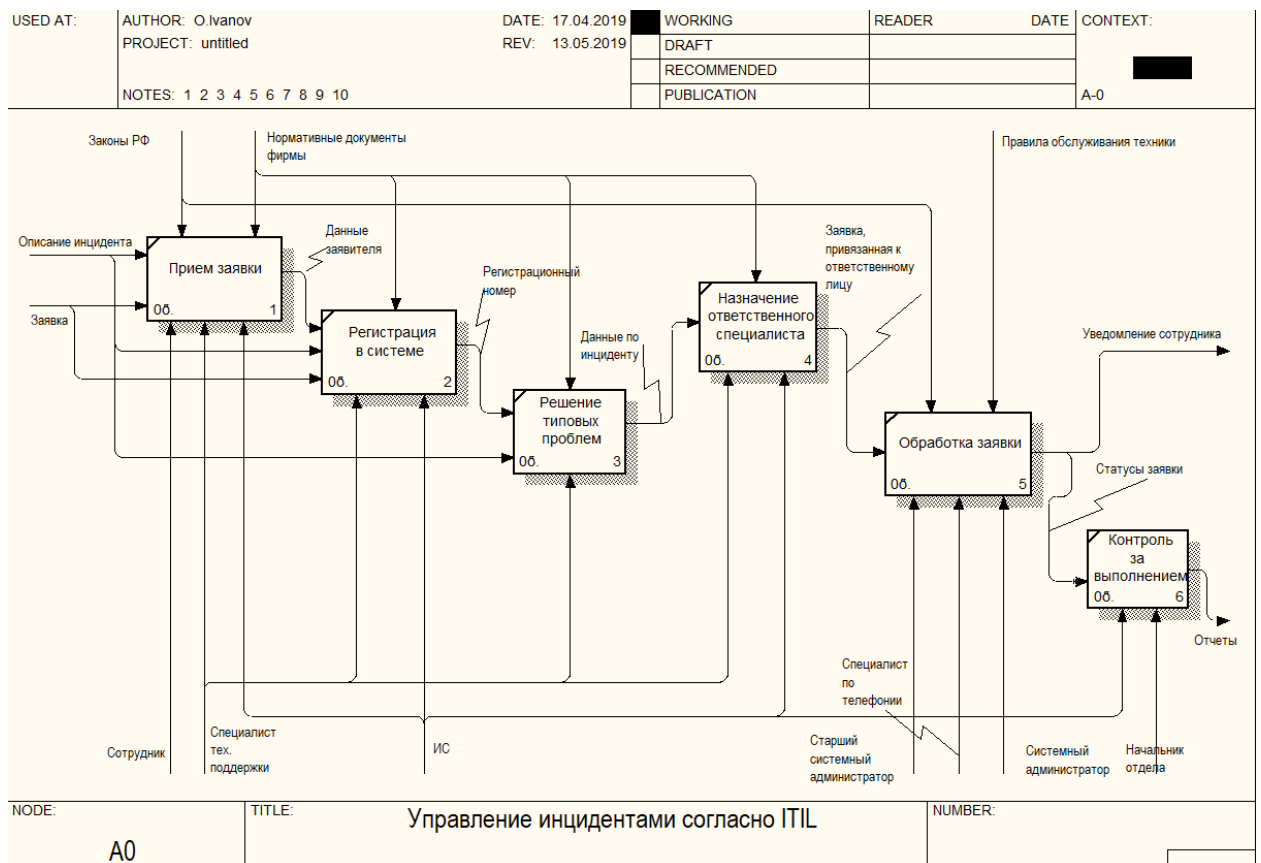


Рисунок 1.5 - Контекстная IDEF0-диаграмма бизнес-процесса управление инцидентами согласно ITIL (1-й уровень)

Улучшение бизнес-процесса управление инцидентами достигается путем внедрения автоматизированной информационной системы ServiceDesk и внедрением первой линии технической поддержки. Все заявки поступают специалисту технической поддержки, который решает простые и типовые заявки ИТ, регистрирует в системе ServiceDesk и назначает ответственного исполнителя. Ответственные исполнители видят назначенные на них заявки и меняют их статусы по мере выполнения. Руководитель департамента ИТ может оперативно увидеть статусы выполняемых заявок или посмотреть отчеты по ним. Внедрение такого подхода позволяет значительно увеличить скорость и качество обработки заявок. Формирование отчетов по выполненным заявкам позволяет найти типовые проблемы и устранить или автоматизировать их решение в дальнейшем.

#### **1.4 Постановка задачи по разработке и внедрению ServiceDesk системы**

##### **1.4.1 Цель и назначение автоматизированного варианта решения задачи**

Цель автоматизации работы специалистов службы технической поддержки – повысить эффективность работы сотрудников службы технической поддержки автомобильной сервисной компании «ООО ЛенАвтоТехника» путем автоматизации их деятельности по обработке и поиску заявок, поступающих от пользователей в информационную службу предприятия.

Задачи, необходимые для достижения поставленной цели:

- описание предметной области и характеристика объекта автоматизации;
- провести описание и анализ процесса по обработке заявок, поступающих от пользователей;
- выполнить обзор существующих систем и выбрать оптимальное решение проблемы;



- выполнить постановку задачи, позволяющей устранить недостатки в процессе обработки заявок, и общую постановку задачи на проектную часть дипломного проекта;
- описать архитектуру разрабатываемой системы;
- выбрать и обосновать методы и подходы, необходимые для проектирования;
- выбрать и обосновать средства, необходимые для проектирования;
- выполнить реинжиниринг процесса учета заявок «Как есть» и построить его функциональную модель «Как будет»;
- с помощью диаграммы прецедентов определить функциональный состав системы;
- на основе диаграммы потоков данных определить существующие хранилища данных, сущности и связи между ними, построить модель «сущность-связь»;
- построить логическую и физическую модели базы данных;
- осуществить проектирование пользовательского интерфейса системы;
- на основе прототипа пользовательских интерфейсов и диаграмм прецедентов составить спецификации прецедентов и диаграммы пригодности;
- на уровне реализации построить диаграммы последовательности и диаграмму классов;
- рассчитать потенциальную экономическую эффективность от проекта.

Использование компьютерной и офисной техники содержит риски, связанные, особенно в случае электронного документооборота, связанные с поломками и отказами техники, которые носят название инциденты.

Процессы обслуживания техники могут быть автоматизированы средствами специального программного обеспечения.

Процессы, которые подлежат автоматизации и управлению средствами ServiceDesk:

1. Ведение проектной документации.
2. Бухгалтерская и финансовая отчетность.

3. Транспортная логистика.
4. Складская логистика.
5. Внутренняя логистика предприятия.
6. Процессы управления.
7. Менеджмент (в т.ч. реклама предприятия и выполняемых работ).

В ходе реализации проекта необходимо:

1. Выполнить бизнес-анализ предметной области, включая:
  - 1.1. Интервьюирование заказчика.
  - 1.2. Анкетирование конечных пользователей.
  - 1.3. Описание заинтересованных лиц и конечных пользователей.
  - 1.4. Словарь предметной области.
  - 1.5. Анализ бизнес-требований (в виде схемы бизнес-целей, бизнес-требований и функций системы).
2. Провести сравнительный анализ существующих аналогичных решений на рынке.
3. Разработать концепцию решения, включая:
  - 3.1. Описание общей функциональности решения.
  - 3.2. Детализация требований к решению (описание пользовательских историй с учетом приоритетов и декомпозиции на подзадачи по методике User Story Mapping).
  - 3.3. Описание границ решения.
  - 3.4. Описание возможных архитектурных и технических решений.
4. Спроектировать разрабатываемую систему, включая:
  - 4.1. Проектирование архитектуры/структуры системы.
  - 4.2. Проектирование базы данных (при необходимости).
  - 4.3. Прототипирование интерфейса пользователя.
  - 4.4. Детальное проектирование отдельных модулей (при необходимости детализации).
5. Выполнить программную реализацию первой версии системы (продукта).

6. Выполнить функциональное тестирование системы по тестовым сценариям.

Система автоматизации учета онлайн заявок позволяет устранить проблемы убрав из цепочки процессов лишнего сотрудника- оператора call-центра, или разгрузив его, что приведет к сокращению количества сотрудников, а значит, к сокращению фонда заработной платы, с другой стороны, строгая форма и рекомендации по ее заполнению (описывались выше) позволят качественно описать проблему, что повысит эффективность работы оператора и качество выполнения заявки, с третьей стороны работа в сети по составлению заявки позволит сократить время подачи заявки и повысит сократит затраты времени сотрудника на подачу заявки.

1.4.2 Формализация моделирования процессов и подпроцессов, реализуемых системой

Система предназначена для автоматизации задач:

1. Авторизация пользователей.
  - a. Авторизация должна проходить по 3 параметрам (логин, пароль и роль).
  - b. При неверно введенных данных, система должна выдавать сообщение о неправильно введенных данных.
2. Разграничение прав пользователей (Администратор, Руководитель, Пользователь, Исполнитель).
  - a. Администратор.
    - i. Отвечает за настройку системы.
      - a) Настройка крайнего срока для заявок.
      - b) Настройка назначения приоритетов в зависимости от пользователя, подающего заявку (организации подразделения должности).
      - c) Настройка сроков для закрытия заявки системой.
    - ii. Управляет пользователями (выдача прав и ролей).
    - iii. Наполняет справочники.

- iv. имеет доступ ко всем объектам системы.
- b. Пользователь может.
  - i. Создавать новые заявки.
  - ii. Просматривать свои заявки, щелкнув по заявке из списка своих заявок.
  - iii. Отправлять заявку на доработку.
  - iv. Закрывать заявки.
  - v. Быстрый поиск по своим заявкам в системе.
  - vi. Фильтрация своих заявок в системе за определенный период по категории, по статусу.
- c. Исполнитель.
  - i. Менять статус заявки на статус «В работе». Если исполнителя взял ее в работу (по нажатию соответствующе кнопки), то статус заявки становится «В работе».
  - ii. Если исполнитель выполнил заявку, то он добавляет решение по ней и ставит заявке статус «Выполнена». Заявка со статусом «Выполнена» направляется руководителю и пользователю, которые могут закрыть ее. Если пользователь, подавший заявку, не закрывает ее со статусом «Выполнена» в определенный срок, то заявку может закрыть руководитель или система. Система должна зафиксировать дату закрытия заявки.
  - iii. Имеет доступ к заявкам, поступившим к нему в работу с возможностью просматривать сведения по заявке по щелчку.
  - iv. Быстрый поиск по своим заявкам в системе.
  - v. Фильтрация своих заявок в системе за определенный период по категории, подкатегории, по статусу, по приоритету, по пользователям.
- d. Руководитель – это исполнитель с флагом руководителя. Он может:
  - i. Создавать Категории и подкатегории заявок (например, категория «оргтехника», подкатегория «Принтера», подраздел «Сканеры», категория «информационные системы», подкатегории «СИР», «ГАСУ», «СМЭВ» и т.д.).

ii. Назначать ответственных сотрудников за ту или иную категорию, подкатеорию, чтобы в дальнейшем заявки, поданные пользователем по данной категории, автоматически переходили на исполнение к назначенному за их исполнение исполнителю).

iii. Менять статус заявки (с «назначена в работу» – на «в работе», с «в работе» - на «выполнена», с «выполнена» на «закрыта», с «выполнена» на «на доработке»). Закрывать заявки.

iv. Добавлять комментарии и решения по заявкам.

v. Менять исполнителей.

vi. Менять приоритет и крайний срок статус заявок.

vii. Осуществлять быстрый поиск среди всех заявок в системе.

viii. Просматривать все заявки системы с возможностью их фильтрации:

a) по заданному периоду (например, с 01.01.2019 по 31.12.2019)

b) по организациям пользователей;

c) по пользователям, подавшим заявки;

d) по категориям и подкатегориям (Категория «Оргтехника»);

e) по типам (инцидент, обслуживание, консультация);

f) по статусам («открыта», «назначена в работу», «в работе», «выполнена», «на доработке» «закрыта»);

g) по приоритетам (высокий, низкий, средний);

h) по исполнителям;

ix. Формировать отчеты.

3. Классификация и автоматическая диспетчеризация входящих заявок. Система должна в зависимости от категории заявки определять исполнителя и отправлять ему входящую заявку на обработку.

4. Отслеживать текущий статус заявки.

5. Формирование отчетности.

Сейчас учет заявок проводится в ручном режиме специалистами департамента ИТ. Но в дальнейшем процесс приема и учета заявок должен

быть автоматизирован, причем автоматизация должна сопровождать весь процесс выполнения заявки.

### **1.5 Анализ существующих разработок и обоснование выбора технологии проектирования**

В ходе обследования предметной области для обзора был определён ряд программных продуктов, реализующих автоматизацию учета и анализа заявок службы технической поддержки. Критериями выбора систем для обзора стали следующие параметры: низкая стоимость, установка на собственный сервер, простота в установке и внедрении, реализация на JavaScript (и jQuery соответственно), PHP использованием стилей bootstrap, СУБД, используемая в продукте должна быть MySQL, только на русском языке, возможность формировать и просматривать отчеты, наличие поиска заявок.

В системе должны быть роли «пользователь» - сотрудник, регистрирующий заявку в системе, «исполнитель» - сотрудник службы технической поддержки, выполняющий работы по заявке, «руководитель» - руководитель службы технической поддержки или заместитель руководителя службы технической поддержки, выполняющие контроль выполнения заявок и формирующие отчетность, «администратор» - сотрудник службы технической поддержки, ответственный за ведение справочников и настройку системы.

Анализ систем, реализующих автоматизацию учета и анализа заявок службы технической поддержки, по выделенным критериям представлен в таблице 1.4.

Таблица 1.4 – Критерии рассматриваемых ServiceDesk систем

<b>Название</b>	<b>Технология</b>	<b>Размещение</b>	<b>Функции</b>	<b>Язык</b>
1	2	3	4	5
Kayako	PHP, Mysql	Частный сервер/SaaS	SLA, портал самообслуживания, live-chat, desktop sharing, мониторинг посетителей, мобильный клиент (Android/IOS)	Английский
Intraservice.ru	PHP, Mysql	Частный	Прием заявок по	Русский

Название	Технология	Размещение	Функции	Язык
		сервер/SaaS	электронной почте, через сайт, учет активов, база знаний, интеграция с Active Directory, 1С, экспорт/импорт Excel	
osTicket	PHP, Mysql	SaaS	Web и email Поддержка, автооповещение, готовые отчеты	Русский, английский
Boas Help Desk	MS SQL, Java, Windows server	SaaS	Учет заявок, быстрый поиск, гибкая система доступа, рейтинги пользователей, отечественная разработка	Русский
VsDesk	PHP, Mysql	Частный сервер (open-source)	Обработка заявок и инцидентов, контроль статусов заявок, отчеты, реестр проблем, база знаний, каталог активов, SLA, интеграция с Active Directory, экспорт/импорт Excel	Русский, английский
Zenlix			SLA, интеграция с Active Directory, уведомления, многоуровневая система прав	Русский, английский

Большинство из рассмотренных систем обладают полным и достаточным функционалом, однако согласно требованиям, следует автоматизировать только процесс по учету и поиску заявок, поэтому приобретение готового решения может привести к дополнительным затратам, что для руководства службы технической поддержки недопустимо.

Как правило, приобретаемые информационные системы долго внедряются и адаптируются под требования заказчиков, что вынуждает часто обращаться в службу поддержки и ведет к дополнительным затратам.

Для учета и сопровождения заявок службой технической поддержки необходимо разработать систему их учета и сопровождения, существующие

системы либо имеют ограниченное количество функций, либо сложны в изучении и внедрении.

### **Выводы по главе 1**

Внедрение информационных систем в работу предприятия содержит риски, связанные с эффективностью системы, ее отказоустойчивостью, помехозащищенностью, гибкостью и ряд других, обусловленных не экономическими, а техническими аспектами, связанными с оборудованием, линиями связи, программным обеспечением. Следует отметить особо риски, связанные с человеческим фактором. Последствия отсутствия надлежащих инструментов для управления информационными технологиями, уровнем технического сопровождения информационных систем, обслуживание компьютерной техники и линий связи может привести к аварийным отключениям системы, что, в свою очередь, может привести к уменьшению производительности, потере доходов и клиентов.

При внедрении на предприятии современных технологий рано или поздно встаёт вопрос о необходимости организации ИТ-службы предприятия и автоматизированной системе Service Desk.



## **Глава 2 Разработка и реализация проектных решений по проектированию, разработке и внедрению service desk в службу технической автомобильной сервисной компании**

### **2.1 Логическое моделирование предметной области**

#### 2.1.1 Логическая модель и ее описание

Алгоритм работы системы:

1. Авторизация. Пользователь входит на страницу авторизации в системе. При помощи логина, пароля и роли проходит авторизацию.

2. Регистрация заявки. Пользователь создает новую заявку, в которой он указывает подкатеорию (при этом категория должна определять автоматически в зависимости от подкатегории), описывает проблему, прикладывает файл. По нажатию на кнопку «ОК» созданной заявке автоматически присваивается статус «Открыта», фиксируется дата и время регистрации заявки в системе. В дальнейшем, пользователь может видеть в списке своих заявок все созданные им заявки. Он может добавлять комментарии по всем незакрытым заявкам. В историю заявки автоматически записывается дата ее создания, описание проблемы и все комментарии, добавленные пользователем. Пользователями могут быть руководитель, исполнитель и администратор.

3. Распределение заявки. Распределение заявок должно осуществляться системой. Система распределяет все заявки, которые не выполнены, по исполнителям в зависимости от подкатегории, указанной в заявке. Если статус заявки «Выполнена» более определенного администратором периода времени система автоматически меняет статус заявки на закрыта. Приоритет заявки определяется в зависимости от сотрудника, подающего заявку (организации подразделения должности), а также от категорий и подкатегорий заявки. Приоритет может изменять руководитель. По умолчанию средний. Крайний срок по умолчанию для всех заявок 14 дней, настраивается администратором.

4. Выполнение заявки. Исполнитель видит все незакрытые заявки, которые назначены им в работу. Если исполнитель имеет флаг руководителя, то ему предоставляются все права руководителя. Если исполнитель без флага руководителя, то он может внести комментарий или решение по заявке и назначить статус заявке «В работе». После проведения работ по заявке исполнитель вносит решение и время, потраченное на работы по заявке, и меняет статус заявки на «Выполнена». Также обязательно указывается тип заявки (консультация, обслуживание, инцидент)

5. Контроль исполнения заявки. Если пользователь не закрывает заявку в течение определенного периода времени (период времени должен настраиваться), то это может сделать руководитель. Также пользователь и руководитель может отправить заявку на доработку, если проблема не устранена.

6. Формирование отчетности. Отчеты имеют форму списка заявок с возможностью фильтрации в зависимости от роли пользователя и статуса заявки. Для каждой роли создается свой список заявок. Пользователь видит все свои заявки по дате от новым к старым и с фильтром по статусу. По умолчанию – все незакрытые заявки за текущий год. Исполнитель видит все заявки, которые назначены ему, с сортировкой по крайнему сроку, приоритету, по дате регистрации от новых к старым и с фильтром по статусу. По умолчанию – все заявки со статусом «Назначена в работу» за текущий год. Руководитель видит все заявки в системе по дате от новым к старым и с фильтром по статусу. По умолчанию – все незакрытые заявки за текущий год. Руководитель может формировать отчеты и просматривать их в графическом виде графиков или диаграмм.

Диаграмма вариантов использования процесса «Управления инцидентами КАК ДОЛЖНО БЫТЬ» представлена на рисунке 2.1.

На диаграмме представлены актеры (действующие лица):

- сотрудник – пользователь, обращающийся в службу технической поддержки (через систему Service Desk);

- специалист технической поддержки – сотрудник, обрабатывающий первую линию пользовательских заявок в техническую поддержку;
- специалист – сотрудник, отвечающий за определенное направление в решении инцидентов (системный администратор, старший системный администратор, специалист по телефонии или руководитель отдела).

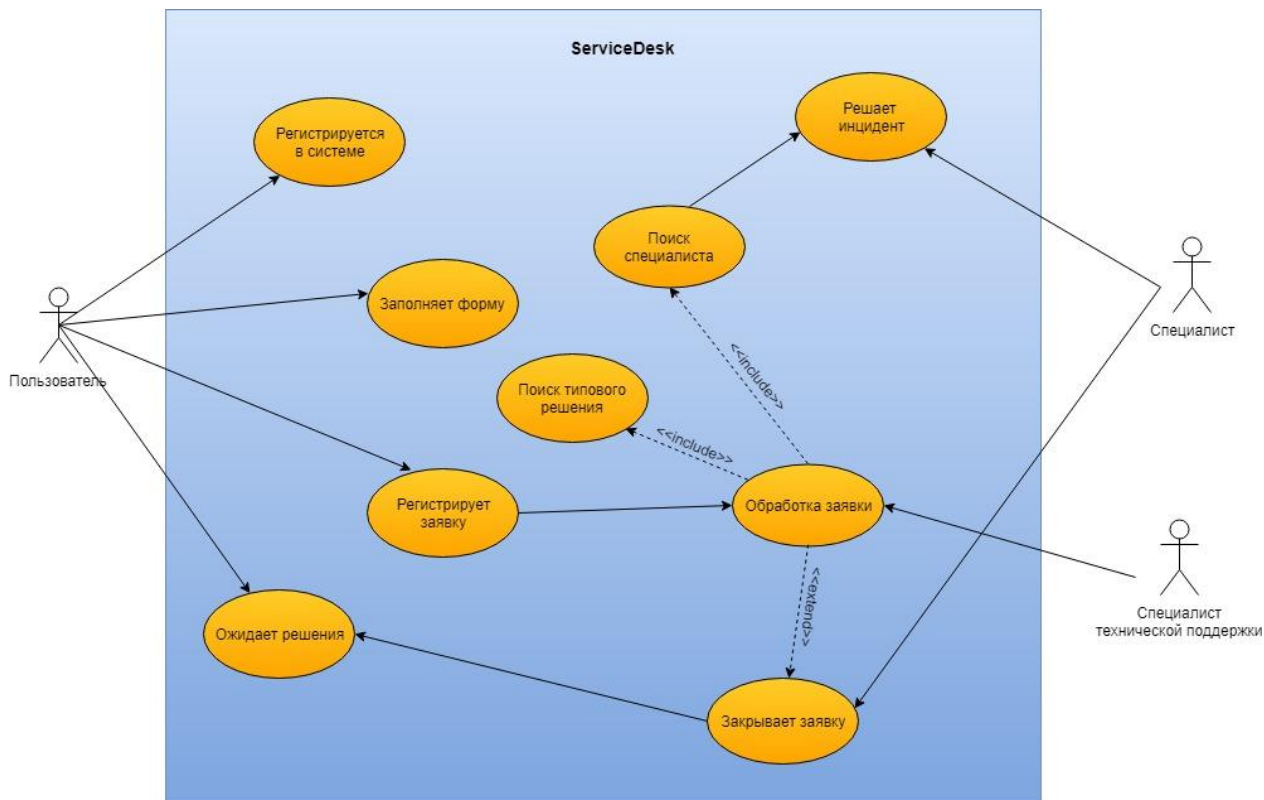


Рисунок 2.1 - Диаграмма вариантов использования бизнес-процесса управления инцидентами «КАК ДОЛЖНО БЫТЬ»

В таблице 2.1 приведем краткую характеристику прецедентов (т.е. самих вариантов использования) диаграммы.

Таблица 2.1 – Характеристика прецедентов

Актер	Прецедент	Характеристика
Сотрудник	Регистрируется в системе	Регистрация пользователя в системе Service Desk
Сотрудник	Заполняет форму	Вносит требуемую информацию по инциденту в систему Service Desk, заполняя поля веб-формы
Сотрудник	Регистрирует заявку	Отправляет заявку по инциденту в службу технической поддержки,

Актер	Прецедент	Характеристика
		через веб-форму Service Desk
Сотрудник	Ожидает решения	Ожидает информацию о решении инцидента, отображение статусов в системе Service Desk;
Специалист службы технической поддержки	Обработка заявки	Обрабатывает поступившие заявки в порядке очереди и в зависимости от указанных приоритетов
Специалист службы технической поддержки	Поиск типового решения	Определяет, возможно ли выполнить заявку типовыми методами (указанными в базе знаний компании)
Специалист службы технической поддержки	Поиск специалиста	Определяет к какому именно специалисту требуется назначить заявку (системному администратору, старшему системному администратору, специалисту по телефонии, руководителю отдела)
Специалист службы технической поддержки	Закрывает заявку	Если инцидент успешно решен без привлечения специалиста, закрывает заявку (изменяет статус на «выполнено»)
Специалист	Решает инцидент	Получает поступившую заявку, осуществляет решение инцидента
Специалист	Закрывает заявку	Закрывает заявку при успешном решении (меняет статус на «закрыто»)

Прежде всего из диаграммы вариантов следует, что система предполагает минимум двух участников непосредственной подачи заявки, это, в первую очередь, сотрудник, который должен заполнить все поля и система, которая должна принять заявку и поместить значения полей в базу данных и сообщить об успешном процессе пользователю (рисунок 2.1).

На рисунке 2.2 представленная диаграмма последовательностей бизнес-процесса управления инцидентами «КАК ДОЛЖНО БЫТЬ». Диаграмма последовательностей показывает взаимодействие объектов во времени.

Пользователь регистрируется в системе ServiceDesk и получает доступ для оформления заявки. Пользователь оформляет и регистрирует заявку об инциденте.

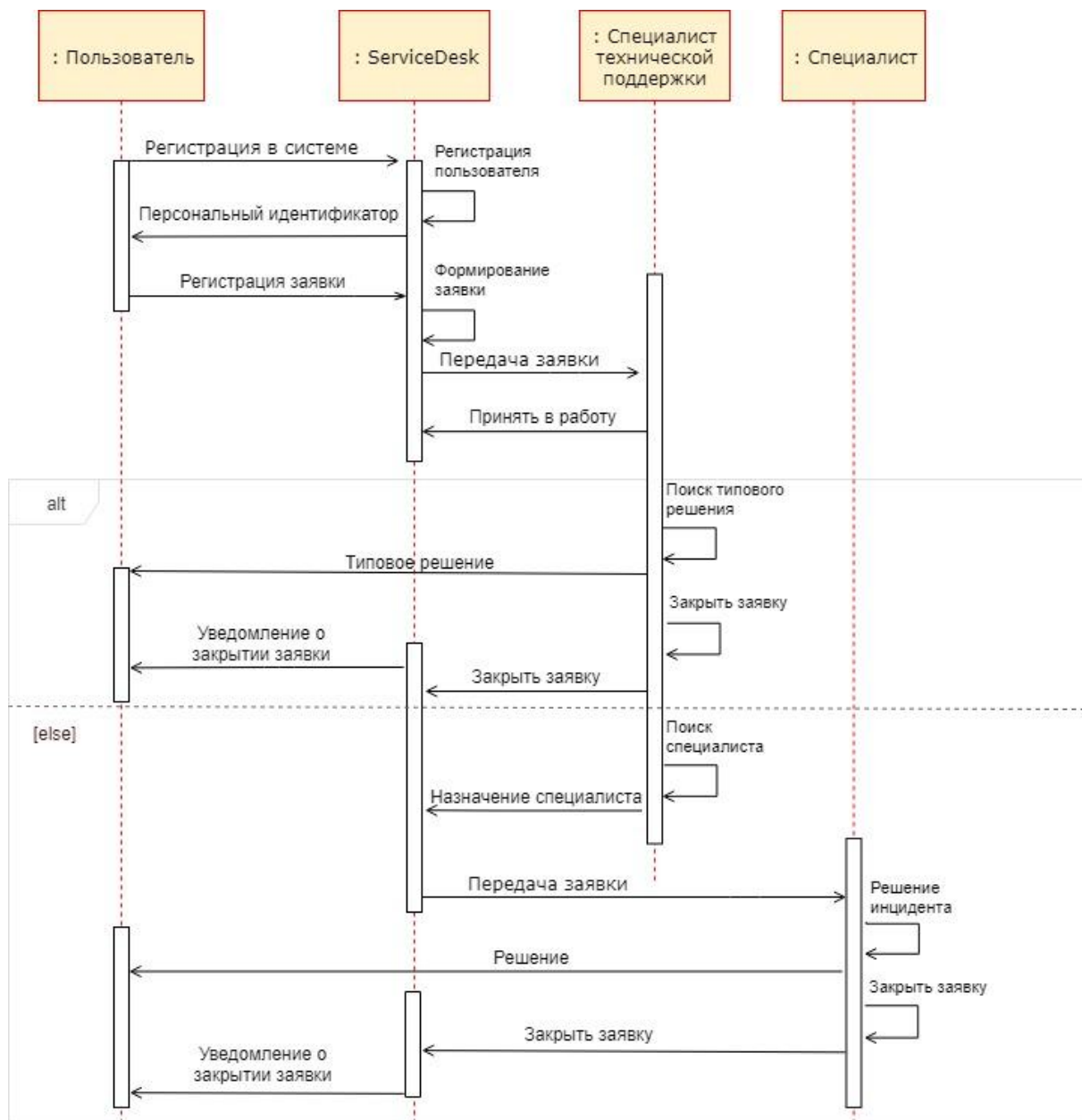


Рисунок 2.2 – Диаграмма последовательности бизнес-процесса управления инцидентами «КАК ДОЛЖНО БЫТЬ»

Специалист технической поддержки видит заявку об инциденте в системе ServiceDesk и принимает ее в работу (меня статус заявки). Если заявка имеет типовое решение, то специалист службы технической поддержки решает и

закрывает заявку. Иначе заявка назначается на соответствующего специалиста, в зависимости от ее категории и проблемы (старший системный администратор, системный администратор, специалист по телефонии или руководитель).

### 2.1.2 Характеристика базы данных

Для построения инфологической модели использовалась нотация IDEF1X, средство проектирования баз данных DBForgeStudio for MySQL (для некоммерческого использования) и база данных MySQL версии 5.7.2.5. Основные сущности диаграммы: клиенты, заявки, прием заявки, сотрудники, список отделов.

Проектирование базы данных осуществляется в три этапа: этап концептуального проектирования, логического проектирования, физического проектирования.

На этапе концептуального проектирования определим сущности и связи между ними, построим модель данных, опираясь на функциональный состав системы.

Итак, в предыдущем разделе мы определили, что в модели данных имеется 13 сущностей.

Исходя из функций системы, первичного документа и хранилищ, выявленных при построении диаграммы потоков данных, выделим сущности рассматриваемой предметной области:

1. Разграничение данных и прав пользователей:
  - a. Учетная запись (Код учетной записи, Логин, Пароль);
  - b. Роль (Код роли, Наименование роли).
2. Регистрация заявок пользователей:
  - a. Пользователи (Код пользователя, Фамилия, Имя, Отчество, Полное имя, Краткое имя, Почта, Телефон);
  - b. Организация (Код организации, Полное наименование, Краткое наименование, ОГРН);
  - c. Подразделение (Код подразделения, Полное наименование подразделения, Краткое наименование подразделения);

- d. Заявка (Код заявки, Дата регистрации, Описание, Файл, Крайний срок);
  - e. Статус (Код статуса, Наименование статуса);
  - f. Подкатегория (Код подкатегории, Наименование подкатегории).
3. Распределение заявок:
- a. Исполнитель (Код исполнителя, Фамилия, Имя, Отчество, Полное имя, Краткое имя, Руководитель);
  - b. Приоритет (Код приоритета, Наименование приоритета);
  - c. Категория (Код категории, наименование категории);
  - d. Должность (Код должности, Наименование должности).
4. Учет времени и решений по выполненным заявкам:
- a. Заявка (Код заявки, Решение, Время решения, Крайний срок);
  - b. Тип заявки (Код типа заявки, Наименование типа заявки).

На основе бизнес правил определим связи между сущностями:

- Каждая учетная запись может принадлежать только одному пользователю.
- Каждая учетная запись может принадлежать только одному исполнителю.
- Каждая учетная запись имеет одну только роль.
- Каждый пользователь работает в одном подразделении.
- Каждый пользователь может сделать одну или несколько заявок.

Каждый пользователь занимает только одну должность.

- Каждое подразделение входит в состав одной организации.
- Каждая должность определяет один приоритет заявки.
- Каждая заявка имеет один приоритет, один статус и один тип.
- В каждой заявке пользователь указывает подкатегорию, категория заявки определяется в зависимости от подкатегории.
- Каждую заявку выполняет один исполнитель.
- За каждую категорию ответственный один исполнитель.
- Каждая категория принадлежит одной категории.

Выделив сущности и определив связи между ними построим ER-модель «Сущность-Связь», представлена на рисунке 2.3.

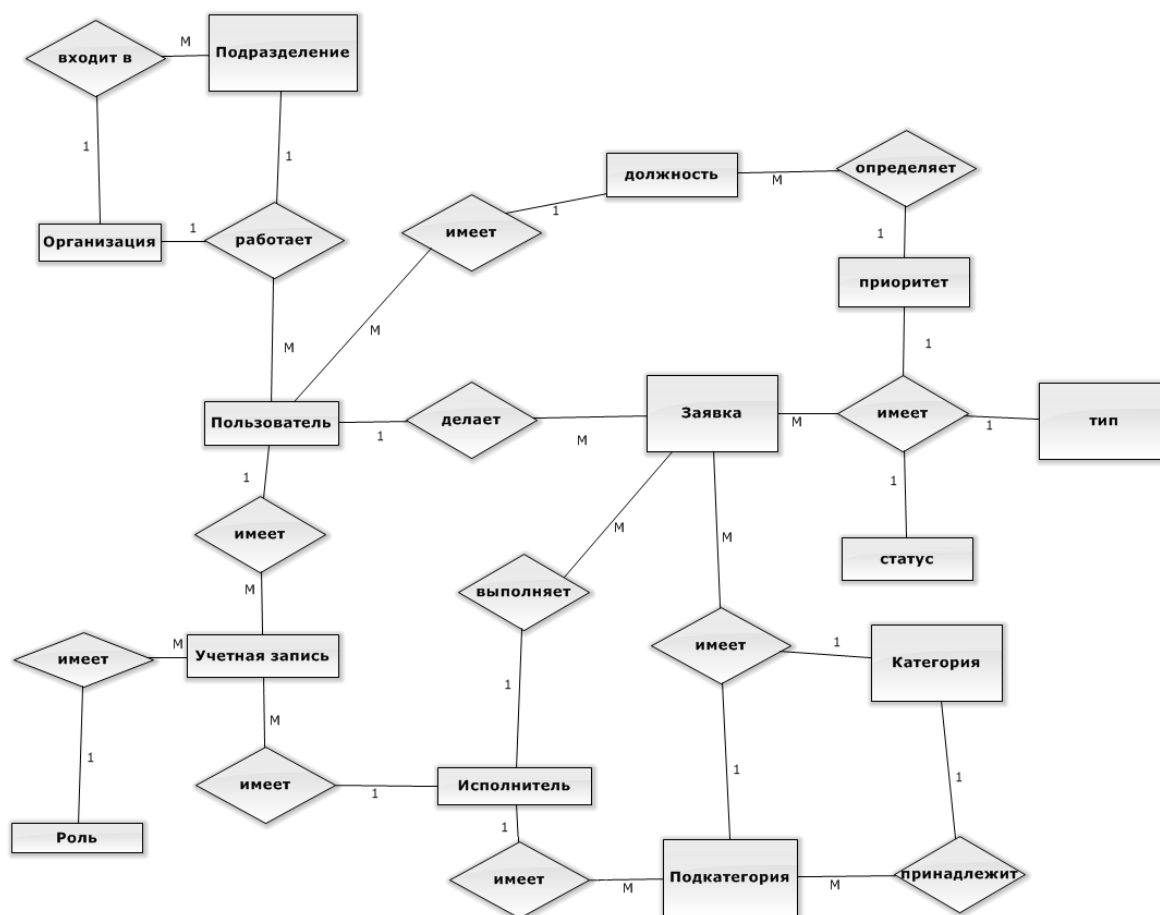


Рисунок 2.3 - ER-модель «Сущность-связь» в нотации Питера Чена

Преобразуем концептуальную модель данных в логическую (инфологическую), воспользовавшись нотацией моделирования IDEF1X. Благодаря инфологической модели мы можем выразить информацию о предметной области, описать объекты, их свойства и взаимосвязи в виде, независимом от системы управления базой данных. В ходе преобразования определим состав атрибутов каждой сущности. Результат преобразования изображен на рисунке 2.4 «Логическая модель базы данных».

На следующем этапе разрабатывается физическая модель данных для СУБД – MySQL Server 5.0.

Разработка физической модели данных основывалась на построенной логической модели данных. Построим диаграммы базы данных.



Проектирование базы данных начинается с того, что выполняем проектирование концептуальной модели.

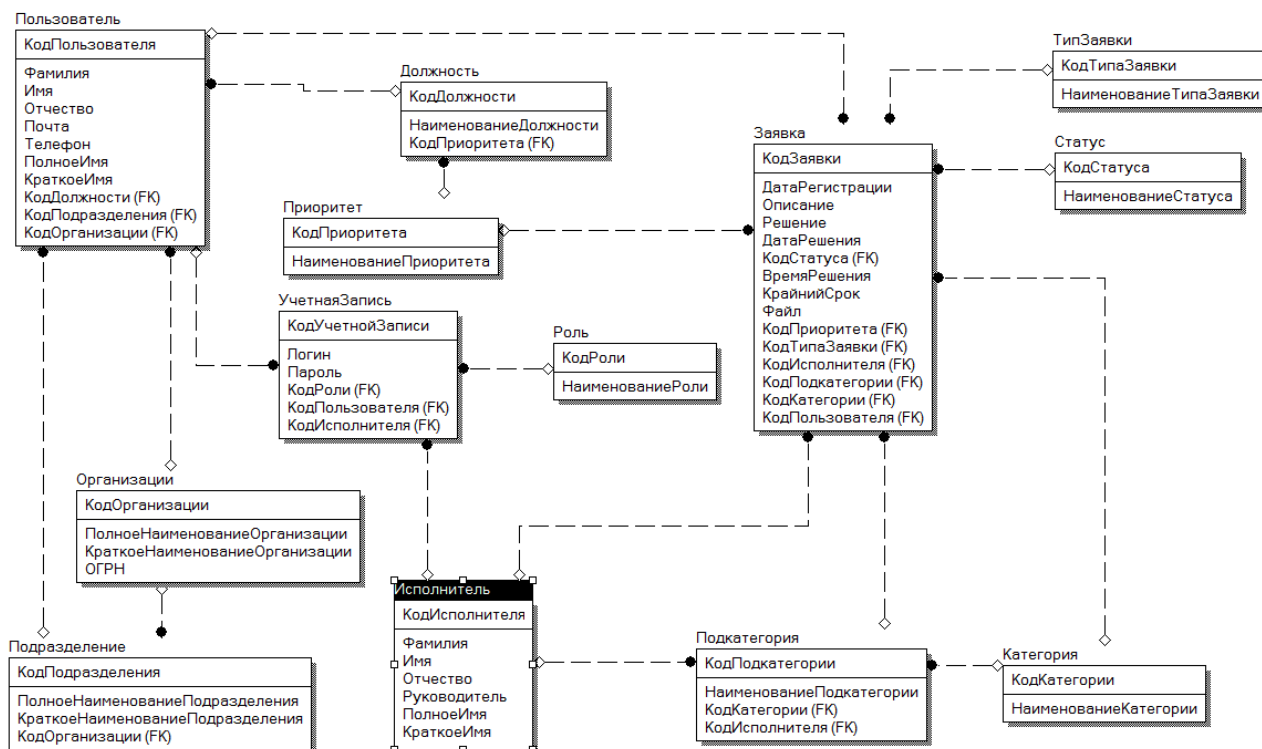


Рисунок 2.4 - Логическая модель базы данных в нотации IDEF1X

Помимо входной оперативной информации в системе используется условно-постоянная информация, которая обычно хранится в виде справочников: категории, подкатегории, типы заявок; статусы заявок, приоритеты заявок, организации, подразделения, должности.

Заполнение справочников должно осуществляться при первичной настройке системы, редактирование – в ходе работы системы. Все операции по работе со справочниками может производить администратор.

### 2.1.3 Характеристика результатной информации

Весь процесс учета заявок пользователей представляет собой систему с входной и выходной информацией.

Входная информация – информация, которую система воспринимает от окружающей среды. Такого рода информация называется входной информацией по отношению к системе.

Входная информация необходима для решения поставленной задачи и может быть расположена на различных традиционных и машинных носителях и представлена в виде документов, сообщений, данных, сигналов, необходимых для выполнения функций системы.

Для решения нашей задачи определим состав видов входной оперативной информации и ее реквизитов.

1. Web-форма ввода сведений по заявке: категория; подкатегория; описание; вложение – файл. Данная форма заполняется каждый раз, когда пользователь (сотрудник органов местного самоуправления) обращается на сервер для ввода содержания и описания, возникшей ошибки.

2. Web-форма авторизации пользователя: логин; пароль; роль. Данная форма заполняется при каждом входе в систему пользователя.

При этом система автоматически идентифицирует пользователя по логину и паролю, подгружая из справочника «Пользователи» информацию о сотруднике. Данные справочника «Пользователи» представлены в таблице 2.2.

Таблица 2.2 Справочник «Пользователи»

Наименование	Тип	Вводящий	Комментарий
Номер	Счётчик	Система	Автоматически
Дата регистрации	Дата и время	Система	Автоматически
Полное наименование	Текст	Администратор	ФИО полностью
Краткое наименование	Текст	Администратор	ФИО в формате Фамилия И. О.
Организация	Ссылка на справочник «Организации»	Администратор	Автоматически в зависимости от пользователя
Подразделение	Ссылка на справочник «Подразделения»	Администратор	Автоматически в зависимости от пользователя
Должность	Текст	Администратор	
Телефон	Текст	Администратор	
Почта	Текст	Администратор	

3. Web-форма ввода сведений по решению заявки: тип; решение; время решения.

Помимо входной оперативной информации в системе используется условно-постоянная информация, которая обычно хранится в виде

справочников: категории; подкатегории; типы заявок; статусы заявок; приоритеты заявок; организации; подразделения.

Заполнение справочников должно осуществляться при первичной настройке системы, редактирование – в ходе работы системы. Все операции по работе со справочниками могут производить администратор.

Выходная информация - информация, получаемая в результате выполнения функций АС и выдаваемая на объект ее деятельности, пользователю или в другие системы при работе информационной системы. Выходная информация может быть представлена в форме данных, выводимых на экран пользователя, и текстовых документов, составленные по запросам пользователя.

По стадиям развития выходную информацию можно подразделить на промежуточную, которая содержит результаты расчётов, используемых в качестве исходных данных для последующих задач. Такая информация обеспечивает связь внутри информационной системы и взаимодействие подсистем. А также выходная информация может быть результатной. Результатная информация - информация, полученная в процессе обработки первичной или промежуточной информации и используемая для управления объектом и принятия решений.

В рамках нашей системы в качестве промежуточной информации будут рассматриваться виды заявок, статус которых меняется в зависимости от выполнения той или иной задачи.

Промежуточная выходная информация:

- зарегистрированная заявка;
- назначенная в работу заявка;
- взятая в работу заявка;
- выполненная заявка;
- отправленная на доработку заявка;
- закрытая заявка.

Результатная информация получается в процессе обработки первичной или промежуточной информации и используется для управления объектом и принятия решений.

В качестве выходной результатной информацией разрабатываемой информационной системы должны быть отчеты:

- отчет по исполнителям за период;
- сводный отчет по исполнителям за период;
- отчет по заявкам в разрезе категорий и подкатегорий, типов за период;
- сводный отчет по заявкам в разрезе категорий, подкатегорий и типов за период;
- сводный отчет по заявкам в разрезе исполнителей и типов за период;
- отчет о выполненных заявках за период;
- отчет о невыполненных заявках за период.

В результате проведенной работы были выявлены основные понятия, связанные с проектированием, классификаторами, справочниками, входной и выходной информации. На основании полученных данных можно приступить к проектированию и разработке системы.

## **2.2 Физическое моделирование АИС**

### **2.2.1 Выбор архитектуры АИС**

В основе разрабатываемой системы лежит архитектура «клиент-сервер», в которой задания или сетевая нагрузка распределены между поставщиками услуг (сервисов), называемых серверами, и заказчиками услуг, называемых клиентами. В качестве среды взаимодействия клиента с сервером используется Интернет.

Основными достоинствами архитектуры «клиент-сервер» является возможность, в большинстве случаев, распределить функции вычислительной системы между несколькими независимыми компьютерами в сети. Это позволяет упростить обслуживание вычислительной системы.

В частности, замена, ремонт, модернизация или перемещение сервера, не затрагивают клиентов - все данные хранятся на сервере, который, как правило, защищён гораздо лучше большинства клиентов.

На сервере проще обеспечить контроль полномочий, чтобы разрешать доступ к данным только клиентам с соответствующими правами доступа - позволяет объединить различные клиенты. Использовать ресурсы одного сервера часто могут клиенты с разными аппаратными платформами, операционными системами.

Основные недостатки:

- в случае использования централизованной системы, неработоспособность основного сервера может сделать неработоспособным все приложение;
- администрирование данной системы требует квалифицированного профессионала;
- высокая стоимость оборудования.

В ходе выбора аппаратной платформы будут предложены и реализованы решения, позволяющие минимизировать вероятность выхода из строя серверной части приложения, а также позволяющее снизить стоимость оборудования до оплаты минимально необходимого уровня производительности.

### 2.2.2 Функциональная схема проекта

Поскольку основной целью приложения является предоставление услуг по средствам сети Интернет, необходимо включить в его серверную часть веб-сервер – аппаратно-программный комплекс, предназначенный для обслуживания HTTP-запросов, которые посылает интернет-браузер клиента. Поскольку приложение предполагает большое число операций по чтению, записи и изменению значительного объема данных, наиболее удобным вариантом будет включение в серверную часть технологий баз данных. Исходя

из выше сказанного серверная часть включает в себя веб-сервер и сервер баз данных, используется трехзвенная архитектура «Клиент-Сервер» (рисунок 2.5).

В задачи веб-сервера входят:

- получение и ответ на HTTP-запросы;
- перенаправление запросов на необходимое приложение (сайт), как правило, приписанное к определенному домену или поддомену;
- предоставление приложениям доступа к необходимым модулям (например, к модулю связи с СУБД, модулю обработки php-программ и другие);
- авторизация и аутентификация пользователей;
- реализация функций файл-сервера.
- 



Рисунок 2.5 – Модель трехзвенной архитектуры «Клиент-Сервер»

В задачи сервера баз данных входит:

- обслуживание запросов на манипуляции с данными на основе языка SQL;
- обеспечение целостности данных;
- предоставление утилит для административного управления СУБД.

Клиентская часть должна состоять из следующих компонент программное обеспечение: операционная система Windows 7 и выше, любой интернет проводник (браузер), поддерживающий: HTML, XHTML, JavaScript, Adobe Flash plug-in, доступ к сети Интернет со скоростью не ниже 512 Кбит\сек.

### 2.2.3 Разработка физической модели данных ServiceDesk

Для разработки физической модели данных использовалось программное обеспечение – MySQL WorkBench 6.3.6.

Результат построения физической модели данных представлен на рисунке 2.6 «Физическая модель базы данных».

Описание таблиц и полей физической модели данных представлены в таблицах 2.3-2.5.

Таблица 2.3. Описание таблиц и полей физической модели данных

Сущность	Атрибут сущности	Ключ	Таблица	Столбец	Тип данных	Обязательное
1	2	3	4	5	6	7
Пользователь	КодПользователя	PK	user	iduser	Int(11)	
	Фамилия			lnameuser	Varchar(20)	
	Имя			fnameuser	Varchar(20)	
	Отчество			snameuser	Varchar(20)	
	Почта			email	Varchar(20)	
	Телефон			phone	Varchar(20)	
	Полное имя			fullnameuser	Varchar(60)	
	Краткое имя			shortnameuser	Varchar(20)	Да
	КодДолжности	FK		idposition	Int(11)	Да
	КодПодразделения	FK		iddepartment	Int(11)	Да
Должность	КодДолжности	PK	position	idposition	Int(11)	Да
	НаименованиеДолжности			nameposition	Varchar(45)	Да
	КодПриоритета	FK		idpriority	Int(11)	Да
Подразделение	КодПодразделения	PK	department	iddepartment	Int(11)	Да
	ПолноеНаименованиеПодразделения			fullnamedep	Varchar(99)	Да
	КраткоеНаименованиеПодразделения			shortnamedep	Varchar(45)	Да
	КодОрганизации	FK		idcompany	Int(11)	Да
Организация	КодОрганизации			idcompany	Int(11)	Да
	ПолноеНаименованиеОрганизации			fullnamecompany	Varchar(99)	Да

Сущность	Атрибут сущности	Ключ	Таблица	Столбец	Тип данных	Обязательное
	КраткоеНаименование Организации			shortnamecompany	Varchar(45)	Да
	ОГРН			ogrn	Varchar(13)	Да
Учетная запись	КодУчетнойЗаписи	PK		idaccount	int(11)	Да
	Логин			login	Varchar(30)	Да
	Пароль			passowrd	Varchar(30)	Да
	КодИсполнителя	FK		idperformer	int(11)	Да
	КодПользователя	FK		iduser	int(11)	Да

Таблица 2.4. Описание таблиц и полей физической модели данных

Сущность	Атрибут сущности	Ключ	Таблица	Столбец	Тип данных	Обязательное
1	2	3	4	5	6	7
Исполнитель	КодРоли	FK		idrole	int(11)	Да
	КодИсполнителя	PK		idperformer		Да
	Фамилия			lperformer	Varchar(20)	
	Имя			fnameperformer	Varchar(20)	
	Отчество			snameperformer	Varchar(20)	
	Полное имя			fullnameperformer	Varchar(60)	
	Краткое имя			shortnameperformer	Varchar(10)	Да
Руководитель			ismanager	boolean	Да	
Категория	КодКатегории	PK		Idcategory	int(11)	Да
	Полное наименование категории			Fullnamcategory	Varchar(30)	Да
	Краткое наименование категории			Shortnamecategory	Varchar(15)	Да
Подкатегория	КодПодкатегории	PK		Idsubcategory	int(11)	Да
	Полное наименование подкатегории			Fullnamsubcategory	Varchar(30)	Да
	Краткое наименование подкатегории			Shortnamesubcategory	Varchar(15)	Да
	КодКатегории	FK		idcategory	int(11)	Да
	КодИсполнителя	FK		idperformer	int(11)	Да



Сущность	Атрибут сущности	Ключ	Таблица	Столбец	Тип данных	Обязательное
Статус	КодСтатуса	PK		idstatus	int(11)	Да
	Наименование статуса			namestatus	Varchar(15)	Да
тип заявк	КодТипа	PK		Idtype	int(11)	Да
	Наименование типа			nametype	Varchar(15)	Да
Роль	КодРоли	PK		Idtype	int(11)	Да
	Наименование роли			nametype	Varchar(15)	Да
Приоритет	КодПриоритета	PK		Idtype	int(11)	Да
	Наименование приоритета			nametype	Varchar(15)	Да

Таблица 2.5. Описание таблиц и полей физической модели данных

Сущность	Атрибут сущности	Ключ	Таблица	Столбец	Тип данных	Обязательное
1	2	3	4	5	6	7
заявка	КодЗаявки	PK		idticket	int(11)	Да
	Дата регистрации			regdate	datetime	Да
	Описание1			comment	Text(999)	Да
	Решение			solution	Text(999)	Да
	ДатаРешения			soldate	datetime	Да
	Крайний срок			duedate	datetime	Да
	Файл			file	File(varchar)	Да
	КодТипаЗаявки	FK		idtype	int(11)	Да
	КодСтатуса	FK		idstatus	int(11)	Да
	КодПодкатегории	FK		idsubcategory	int(11)	Да
	КодКатегории	FK		idcategory	int(11)	Да
	КодИсполнителя	FK		idperformer	int(11)	Да
КодПриоритета	FK		idpriority	int(11)	Да	

Сущность	Атрибут сущности	Ключ	Таблица	Столбец	Тип данных	Обязательное
	КодПользователя	FK		iduser	int(11)	Да

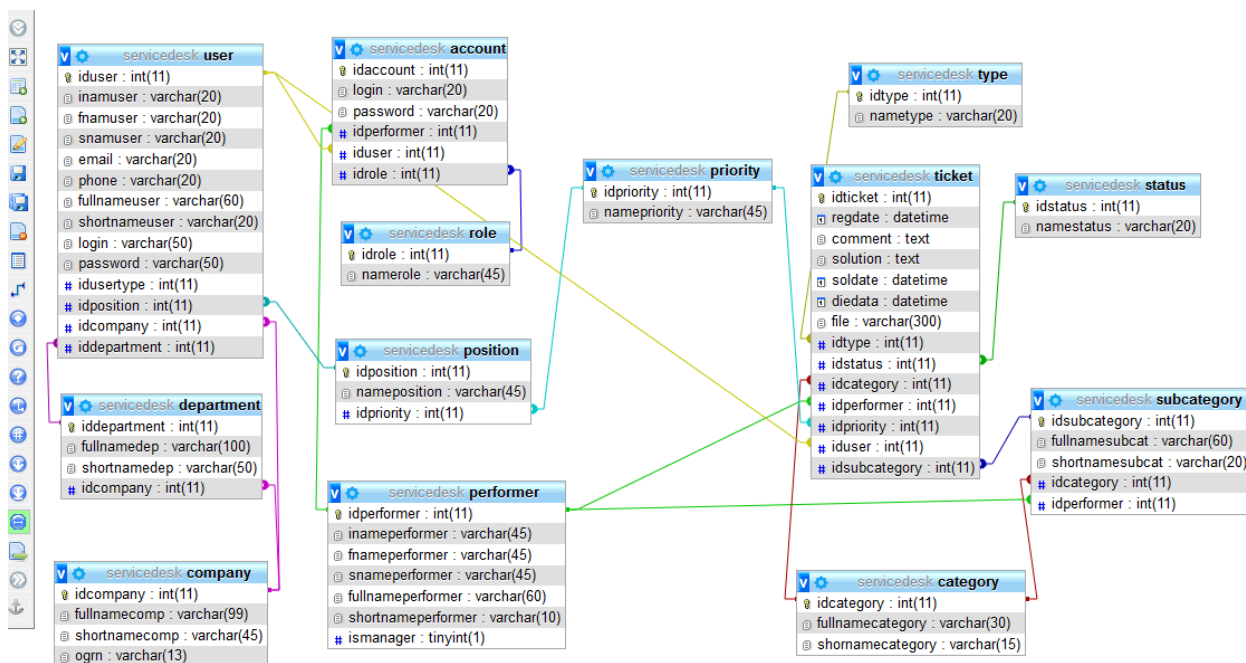


Рисунок 2.6 - Физическая модель данных

Физическая модель данных представляет тринадцать таблиц: «user», «department», «company», «account», «role», «position», «performer», «priority», «type», «ticket», «category», «status», «subcategory». Атрибуты таблиц имеют определенные типы данных. Таблицы связываются между собой при помощи первичного и внешнего ключей.

#### 2.2.4 Структурная схема проекта

Для выявления всех необходимых функций Service Desk системы по обработке заявки, ее сопровождению и хранению, рассмотрим работу подразделения «Служба технической поддержки предприятия», где работа с заявками наиболее характерно выражена.

Основные операции службы технической поддержки предприятия ООО «ЛенАвтоТехника»:

- прием, регистрация заявок от пользователей;

- идентификация и обработка заявок (регистрация инцидентов) на обслуживание;
- учет информации по решенным проблемам;
- информирование пользователей о текущем статусе обращений;
- контроль сроков исполнения заявок;
- передача заявок специалистам более высокой квалификации;
- информирование пользователей о проведении плановых работ, изменений и т.д.

Служба технической поддержки ООО «ЛенАвтоТехника» обеспечивает:

- единую точку контакта к службе поддержки. Удобный и понятный механизм подачи заявки на обслуживание (решение инцидента) позволит более быстро решать их проблемы;
- стандартный способ регистрации заявок;
- определение уровня заявки, подбора специалиста и переадресация заявки, при необходимости на второй уровень, выдачи заданий специалистам;
- контроль над последовательностью выполненных работ, потраченного времени и ресурсов;
- назначение приоритетов запросам в зависимости от типа запроса, конкретного клиента или других обстоятельств;
- хранение базы знаний по решенным проблемам, типичным и характерным неисправностям, позволяющее специалистам быстро решать проблемы, схожие с уже возникавшими.

При реализации системы on-line автоматической регистрации в службе технической поддержки предприятия снижается вероятность возникновения «очередей» заявок у высококвалифицированных ИТ-специалистов, которые будут совершенно неэффективно тратить свои ресурсы. Место в структуре предприятия системы Service Desk показано на рисунке 2.7. Сейчас, система служба технической поддержки (Service Desk), позволяет спустя некоторое время, определенное регламентом вида работ, разрешать до 65% заявок еще на

первой линии. Это возможно благодаря существующей базе знаний и другим преимуществам правильно организованных процессов.



Рисунок 2.7 - Место Service Desk в структуре предприятия

Использование on-line регистрации позволит повысить этот показатель до 70%, за счет сокращения времени обработки заявки. Оставшиеся 30% заявок маршрутизируются и передаются на второй уровень с привлечением специалистов узкого профиля, что гарантирует их скорейшее решение.

### 2.2.5 Описание программных модулей

В службу поддержки входят следующие блоки: диспетчер службы (специалист службы технической поддержки), специалисты службы, управляющий менеджер службы. В ITIL служба поддержки реализует процесс «Управление инцидентами». Его образуют функции: прием и регистрация инцидента, первоначальная классификация и поддержка, исследование и диагностика, решение и восстановление, закрытие инцидента. За выполнение первых двух этапов отвечает диспетчер службы. Третий и четвертый этапы возложены на специалистов службы. И, наконец, пятый этап возложен на диспетчера службы. Допускается, что решение о закрытии инцидента принимает управляющий менеджер службы. В его обязанности входит анализ работы ИТ-инфраструктуры предприятия, основанный на данных об

инцидентах. Часто это бывает поиск «узких» мест действующей системы (например, ненадежное оборудование, слабая подготовка пользователей, устаревший парк техники).

Служба технической поддержки организована по многоуровневому принципу (рисунок 2.8):

- Пользователь – обращается с вопросом в службу поддержки по телефону или, после введения разработанной системы, с помощью электронной заявки через модуль автоматизированной on-line службы технической поддержки предприятия.
- Специалист технической поддержки – регистрирует обращение, при возможности помогает пользователю самостоятельно, либо передаёт заявку на вторую линию и контролирует ее выполнение.
- Вторая линия поддержки (системный администратор, старший системный администратор, специалист по телефонии) – получает заявки от первой линии, и выполняет работы по ним.

Алгоритм обработки заявки службой технической поддержки показан на рисунке 2.9.

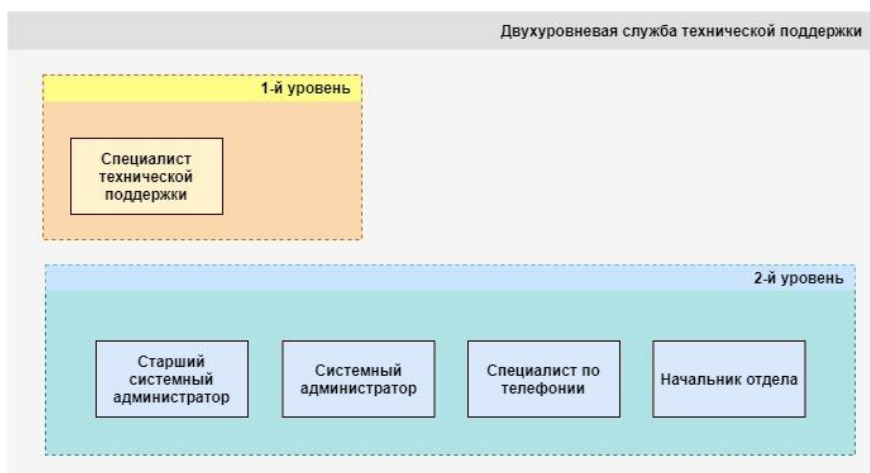


Рисунок 2.8 - Структура двухуровневой службы технической поддержки

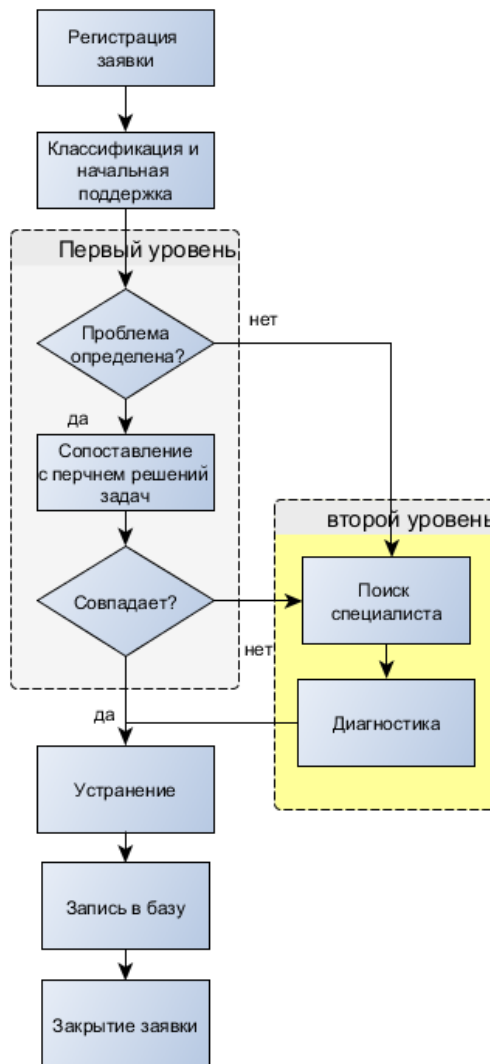


Рисунок 2.9 - Алгоритм обработки заявки службой технической поддержки

На основании выделенных функций по работе с заявками в службе ИТ предприятия можно рассмотреть общий алгоритм обработки и сопровождения заявки оператором системы *Serves Desk* любой службы или подразделения предприятия. Следует обратить внимание на следующие моменты:

- заявка вводится в ручном режиме;
- система предполагает наличие специалиста технической поддержки, который назначает специалиста на выполнение заявки, в зависимости от подразделения (назовем такого специалиста в каждом подразделении оператором системы);
- система предполагает закрытие заявки специалистом, который ее выполнял.

В данной работе основное внимание будет уделяться вопросу обработки заявок, т.е. второй составляющей, так как первая не предполагает учета и регистрации запросов клиентов.

## 2.2.6 Схема взаимосвязи программных модулей и информационных файлов

Система строится на основе архитектурного паттерна MVC. Модель-вид-контроллер (или модель-представление-контроллер, англ. Model-view-controller, MVC) - архитектурный шаблон, который используется при проектировании и разработке программного обеспечения. Этот шаблон предполагает разделение системы на три взаимосвязанные части: модель данных, вид (интерфейс) и модуль управления. Применяется для отделения данных (модели) от интерфейса (вида) так, чтобы изменения интерфейса минимально влияли на работу с данными, а изменения в модели данных могли осуществляться без изменений интерфейса. Цель шаблона - гибкий дизайн программного обеспечения, который должен облегчать дальнейшие изменения или расширения программ, а также предоставлять возможность повторного использования отдельных компонентов программы. Кроме того, использование этого шаблона в больших системах способствует упорядоченности их структуры и делает их более понятными за счет уменьшения сложности.

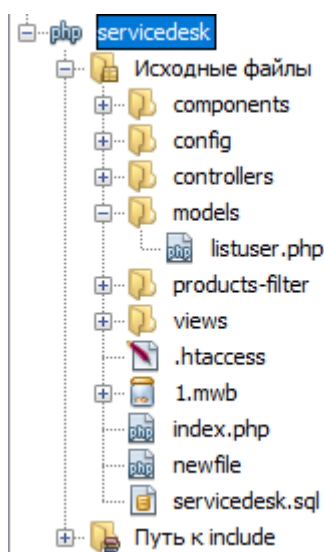


Рисунок 2.10 - Структура проекта на основе MVC шаблона

В рамках архитектурного шаблона модель-вид-контроллер (MVC) программа делится на три отдельные, но взаимосвязанные части с распределением функций между компонентами. Структура проекта на основе MVC показана на рисунке 2.10. Модель (Model) отвечает за хранение данных и обеспечения интерфейса к ним. Вид (View) ответственный за представление этих данных пользователю. Контроллер (Controller) руководит компонентами, получает сигналы в виде реакции на действия пользователя (изменение положения курсора мыши, нажатия кнопки, вот данных в текстовое поле) и передает данные в модель. Модель является центральным компонентом шаблона MVC и отражает поведение приложения, независимую от интерфейса. Модель касается прямого управления данными, логикой и правилами приложения.

Вид может представлять собой любое представление информации, получаемое на выходе, например, график или диаграмму. Одновременно могут сосуществовать несколько выгядел (представлений) одной и той же информации, например, гистограмма для руководства компании и таблицы для бухгалтерии. Контроллер получает входные данные и преобразует их в команды для модели или вида. Модель инкапсулирует ядро – данных и основной функционал их обработки и не зависит от процесса ввода или вывода данных. Может выгядет несколько взаимосвязанных областей, например, различные таблицы и поля форм, в которых отражаются данные.

В функции контроллера входит отслеживание определенных событий, возникающих в результате действий пользователя. Контроллер позволяет структурировать код путем группирования связанных действий в отдельный класс. Например, в типичном MVC-проекте может быть пользовательский контроллер, содержит группу методов, связанных с управлением учетной записью пользователя, таких как регистрация, авторизация, редактирование профиля и изменение пароля. Таким образом, основная нагрузка бизнес-логики приложения реализуется именно файлами контроллера (рисунок 2.11).



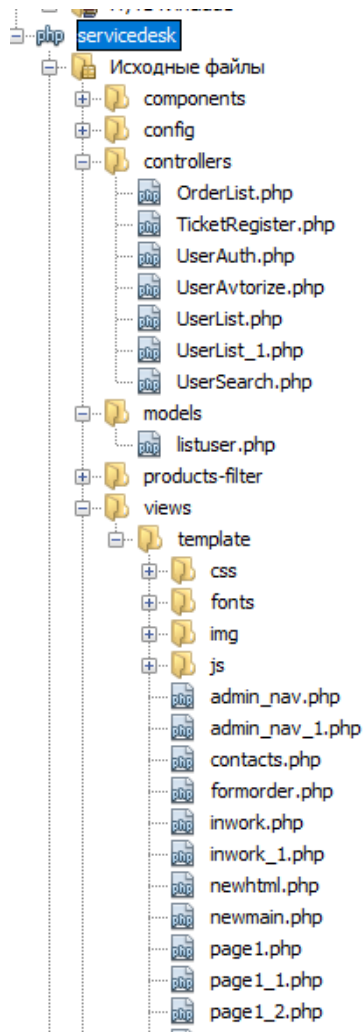


Рисунок 2.11 - Структура модели, просмотрщика и контроллера

Модель содержит механизм обработки данных, которые поступают из соответствующего контроллера:

```
<html>
  <head>
    <meta charset="UTF-8">
    <title></title>
  </head>
  <body>
    <?php
      include_once $_SERVE
```

```
R['DOCUMENT_ROOT'].'servicedesk/controllers/UserList.php';
```

```
    $use=new UserList();
    echo $use->getUsersList();
  ?>
```

```
</body>
</html>
```

Контроллеры содержат соответствующие функции, например получение данных пользователей OrderList.php:

```
<!DOCTYPE html>
<html>
  <head>
    <title> ПК конструктор - Главная </title>
    <link rel="stylesheet" href="bootstrap\style.css" type="text/css">
    <link rel="stylesheet" href="bootstrap/css/bootstrap.css" type="text/css">
    <meta charset="utf-8">
  </head>
  <body>
    <?php
    //session_start();
    require_once dirname(__DIR__).'components/Db.php';

    class NewOrderList {

      public static function getOrderItemById($id)
      {
        $id=intval($id);
        if($id){
          $db=Db::getConnection();
          $result=$db->query('SELECT*from ticket WHERE id='. $id);
          $result->setFetchMode(PDO::FETCH_ASSOC);

          $newsItem=$result->fetch();
        }
      }

      public static function getOrderList()
      {
        $db=Db::getConnection();
        $newList=array();
        //if($db){echo 'connect';}else{echo 'connect error';}
        // result=$db->query('SELECT idticket, regdate, comment, solution, soldate, diedata,
file, idtype, idstatus, idcategory, idperformer, idpriority, iduser, idcategory'
        $result=$db->query('SELECT * FROM ticket ORDER BY idstatus DESC LIMIT 10');
        $i=0;
        $row=array();
        while($row=$result->fetch()){
          $newList[$i]['idticket']=$row['idticket'];
          $newList[$i]['regdate']=$row['regdate'];
          $newList[$i]['comment']=$row['comment'];
          $newList[$i]['solution']=$row['solution'];
          $newList[$i]['soldate']=$row['soldate'];
          $newList[$i]['diedata']=$row['diedata'];
```

```

        $newList[$i]['file']=$row['file'];
        $newList[$i]['idtype']=$row['idtype'];
        $newList[$i]['idstatus']=$row['idstatus'];
        $newList[$i]['idcategory']=$row['idcategory'];
        $newList[$i]['idperformer']=$row['idperformer'];
        $newList[$i]['idpriority']=$row['idpriority'];
        $newList[$i]['iduser']=$row['iduser'];
        $newList[$i]['idsubcategory']=$row['idsubcategory'];
        $i++;
    }
    return $newList;
}
} ?>
</body>
</html>

```

Или обработка заявки TicketRegister:

```

<?php
require_once dirname(__DIR__).'/components/Db.php';

class TicketRegister {
    function __construct(){}

    public function Set_Ticket(){
        $db=Db::getConnection();
        if($db) echo 'db';
        $newList=array();
        result=$db->query('SELECT idticket, regdate, comment, solution, soldate, diedata, file,
idtype, idstatus, idcategory, idperformer, idpriority, iduser, idcategory'
        $query='INSERT INTO ticket (iduser, regdate, comment, file, idtype, idstatus, idcategory,
idsubcategory, idpriority)
            VALUES(:iduser, :regdate, :comment, :file, :idtype, :idstatus, :idcategory,
:idsubcategory, :idpriority)';
        $stmt = $db->prepare($query);
        $stmt->bindParam(':iduser', $_POST['iduser'], PDO::PARAM_STR);
        $stmt->bindParam(':regdate', $_POST['regdate'], PDO::PARAM_STR);
        $stmt->bindParam(':comment', $_POST['comment'], PDO::PARAM_STR);
        $stmt->bindParam(':file', $_POST['file'], PDO::PARAM_STR);
        $stmt->bindParam(':idtype', $_POST['idtype'], PDO::PARAM_STR);
        $stmt->bindParam(':idstatus', $_POST['idstatus'], PDO::PARAM_STR);
        $stmt->bindParam(':idcategory', $_POST['idcategory'], PDO::PARAM_STR);
        $stmt->bindParam(':idsubcategory', $_POST['idsubcategory'], PDO::PARAM_STR);
        $stmt->bindParam(':idpriority', $_POST['idpriority'], PDO::PARAM_STR);
        $stmt->execute();
    }
}
}

```

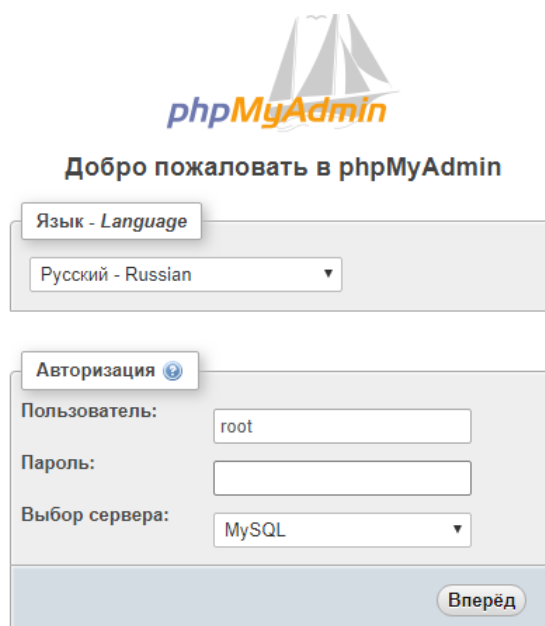
Просмотрщик содержит модель шаблона и вид отображения отдельных страниц. Пример такой страницы приведен в приложении Б.

### 2.3 Контрольный пример реализации проекта и его описание

Для проверки работоспособности разработанной системы Service Desk был выполнен ряд контрольных заданий.

1. Установка системы на компьютер. Для сетевой версии – разворачиваем систему на веб сервере, базу данных переносим на сервер баз данных. В папке конфигурации системы прописывает значения имени, пользователя и паролей доступа к базе данных. Часть данных является константами и прописывается в файле const.php.

2. Если система устанавливается для тестирования на локальный компьютер, то сначала устанавливаем локальный веб сервер и сервер базы данных, рекомендуется скачивание и установка веб платформы XAMPP с версией языка программирования 7.2.17. В пакетах XAMP запускаем серверы Apache и MySQL. Файлы проекта из архива (servicedesk.zip) распаковываем в папку htdocs. Для работы проекта создаем базу данных и ее структуру, для этого запускаем phpMyAdmin (в адресной строке браузера набираем localhost и в выпавшем меню ищем phpMyAdmin, указываем логин root и пустой пароль). Процесс авторизации указан на рисунке 2.12.



phpMyAdmin

Добро пожаловать в phpMyAdmin

Язык - Language

Русский - Russian

Авторизация

Пользователь: root

Пароль:

Выбор сервера: MySQL

Вперёд

Рисунок 2.12 - Авторизация на сервере баз данных MySQL

Процесс создания пустой базы данных с названием servicedesk указан на рисунке 2.13.

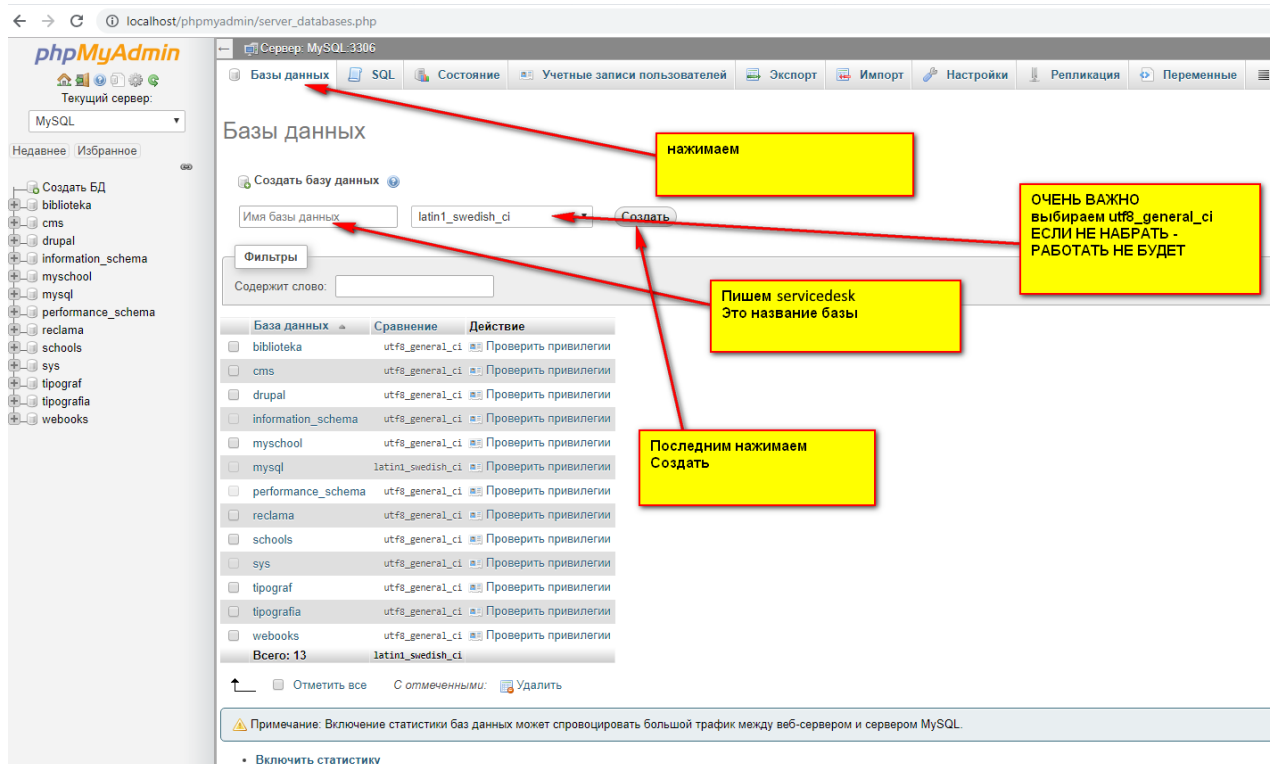


Рисунок 2.13 - Создание базы данных, в нашем случае servicedesk

Импорт структуры базы данных из файла servicedesk.sql представлен на рисунках 2.14-2.15.

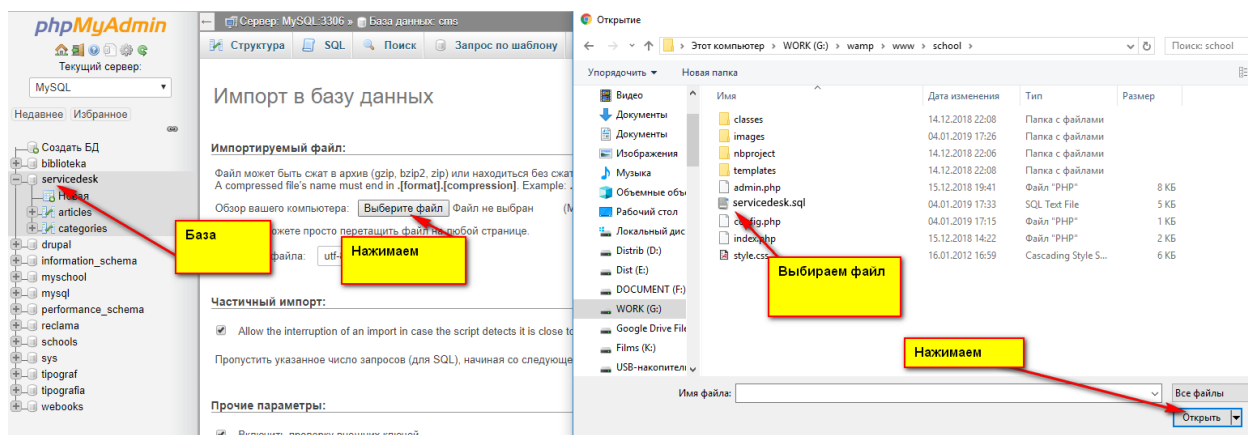


Рисунок 2.14 - Импорт базы данных из файла servicedesk.sql

Входим в систему в режиме администратора, попадаем в панель администратора (логин: admin, пароль:123).

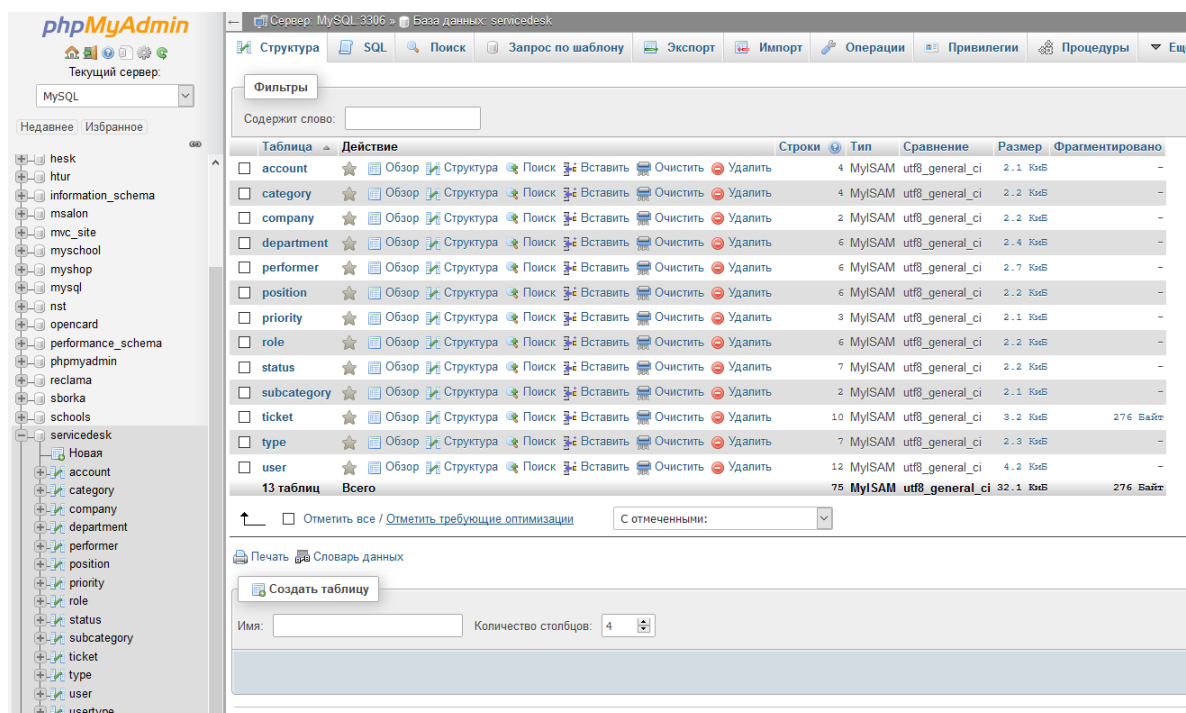


Рисунок 2.15 - Результат импорта базы данных

3. Для того, чтобы попасть на главную страницу servicedesk, в любом браузере запускаем localhost/servicedesk/. Главная страница показана на рисунке 2.16.

Панель администратора системы и интерфейс регистрации нового пользователя показан на рисунке 2.17.

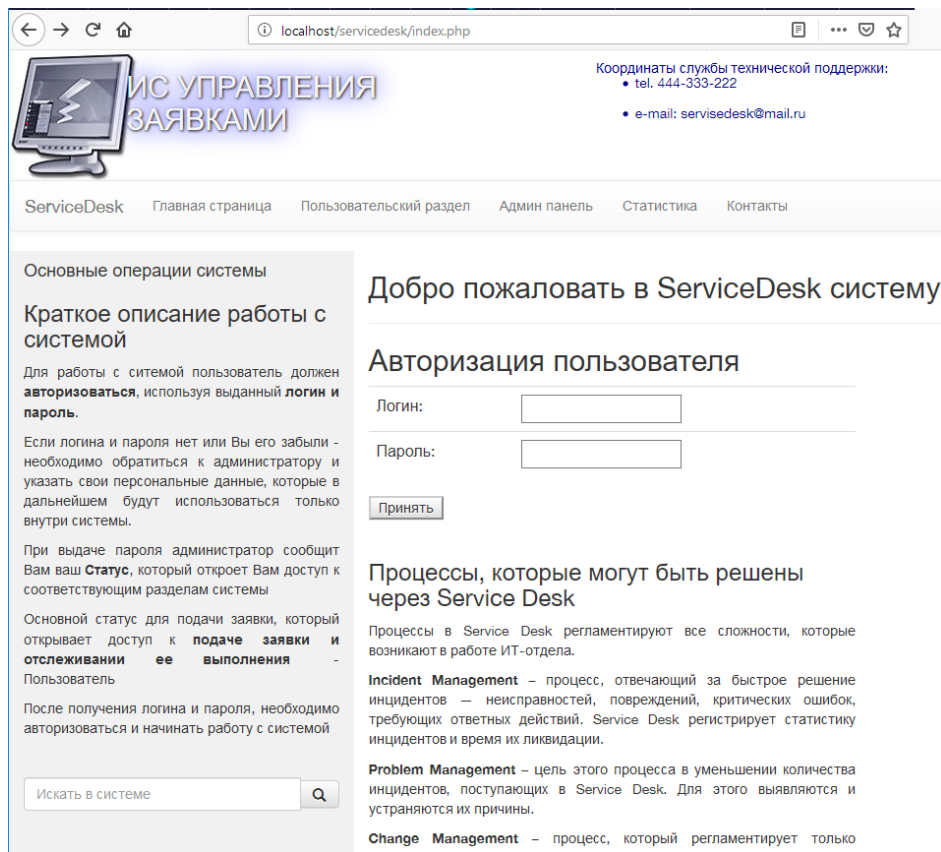


Рисунок 2.16 - Главная страница системы

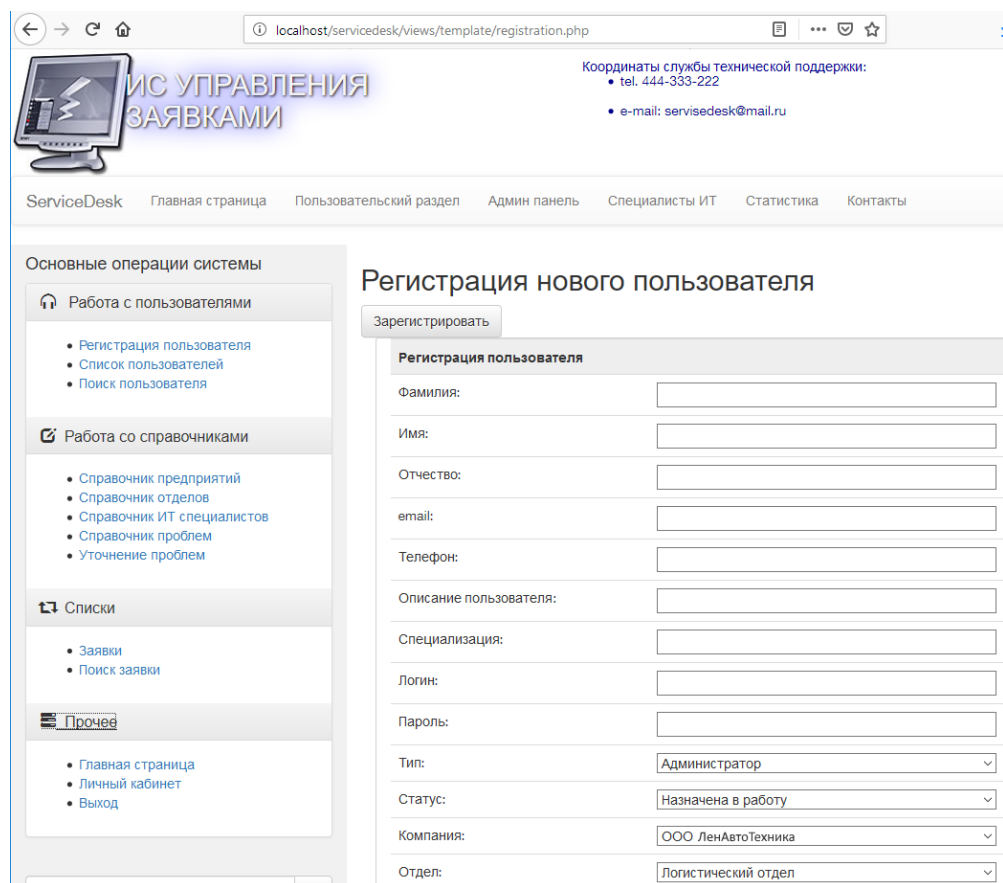


Рисунок 2.17 - Главная страница панели администратора - регистрация нового пользователя

В панели администратора доступен ряд операций, скрытых в левом меню и доступные администратору:

- работа с пользователями, регистрация, просмотр, поиск (рисунок 2.18);
- работа со справочниками: предприятий, отделов, специалистов;
- списки, обработка заявок,
- прочее, переходы.

Для администратора активным является только раздел работы с пользователями – остальные разделы пассивны, но при необходимости они могут быть активизированы.

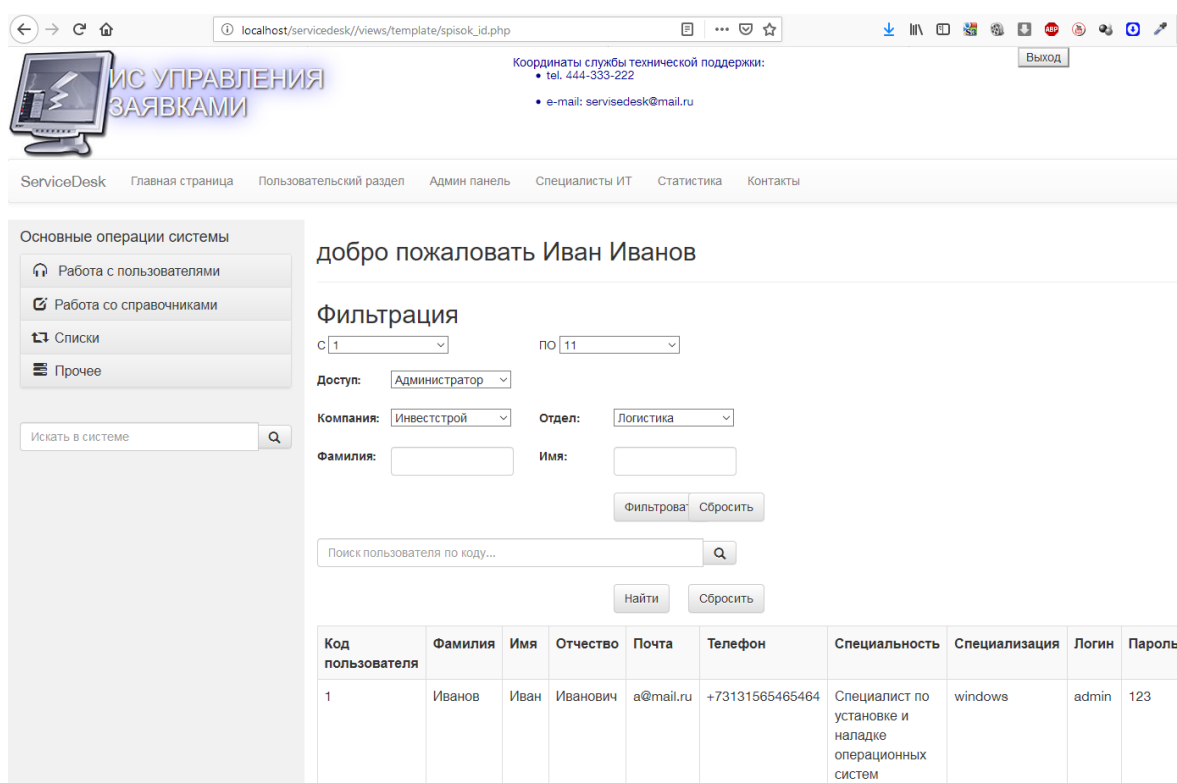


Рисунок 2.18 - Поиск в системе

Принцип простой администратор не участвует в процессе создания и обработки заявок. Если выйти из системы, то можно зайти, как специалист (логины Spes1, Spes2 и пароли 123).

Для специалиста службы технической поддержки доступны следующие функции (рисунок 2.19):



- новое обращение;
- создать стандартный запрос;
- оценка качества ИТ;
- мои обращения: поиск, просмотр открытых, обращения для подтверждения, отклоненные обращения;
- мои согласования: активные согласования, архив согласований;
- прочее: главная страница, кабинет, выход.

Основные операции системы

добро пожаловать Петр Петров

Фильтрация

с 1 по 1

Категория: 1 Статус: 1

Подкатегория: 1 Приоритет: 1

Пользователь: 1 Тип: 1

Фильтры Сбросить

Поиск заявки...

Найти Сбросить

Код заявки	Дата регистрации	Проблема	Решение	Дата выполнения	Дата закрытия	Файл	Тип	Статус	Категория	Подкатегория	Код исполнителя
2	2018-03-25 18:45:45	Нет подачи бумаги					1	1	1	1	
3	2018-03-25 20:02:15	Нет подачи бумаги					1	1	1	1	
4	2018-03-25 20:24:51	Шум в системном блоке					3	1	1	1	
5	2018-03-25 20:31:57	Не работает компьютер					3	1	1	1	
6	2018-03-25 20:33:57	Не работает компьютер					3	1	1	1	

Рисунок 2.19 - Страница специалиста службы технической поддержки

На страницу пользователя можно зайти, указав логин (user2, user3) и пароль пользователя (123). Функции:

- новое обращение;
- создать стандартный запрос;
- оценка качества ИТ;
- мои обращения: поиск, просмотр открытых, обращения для подтверждения, отклоненные обращения;
- мои согласования: активные согласования, архив согласований;
- прочее: главная страница, кабинет, выход.

Для руководителя службы тех поддержки в системе servicedesk представлен свой раздел, для входа логин boss1 или boss2, пароль 123. Интерфейс для руководителя показан на рисунке 2.20.

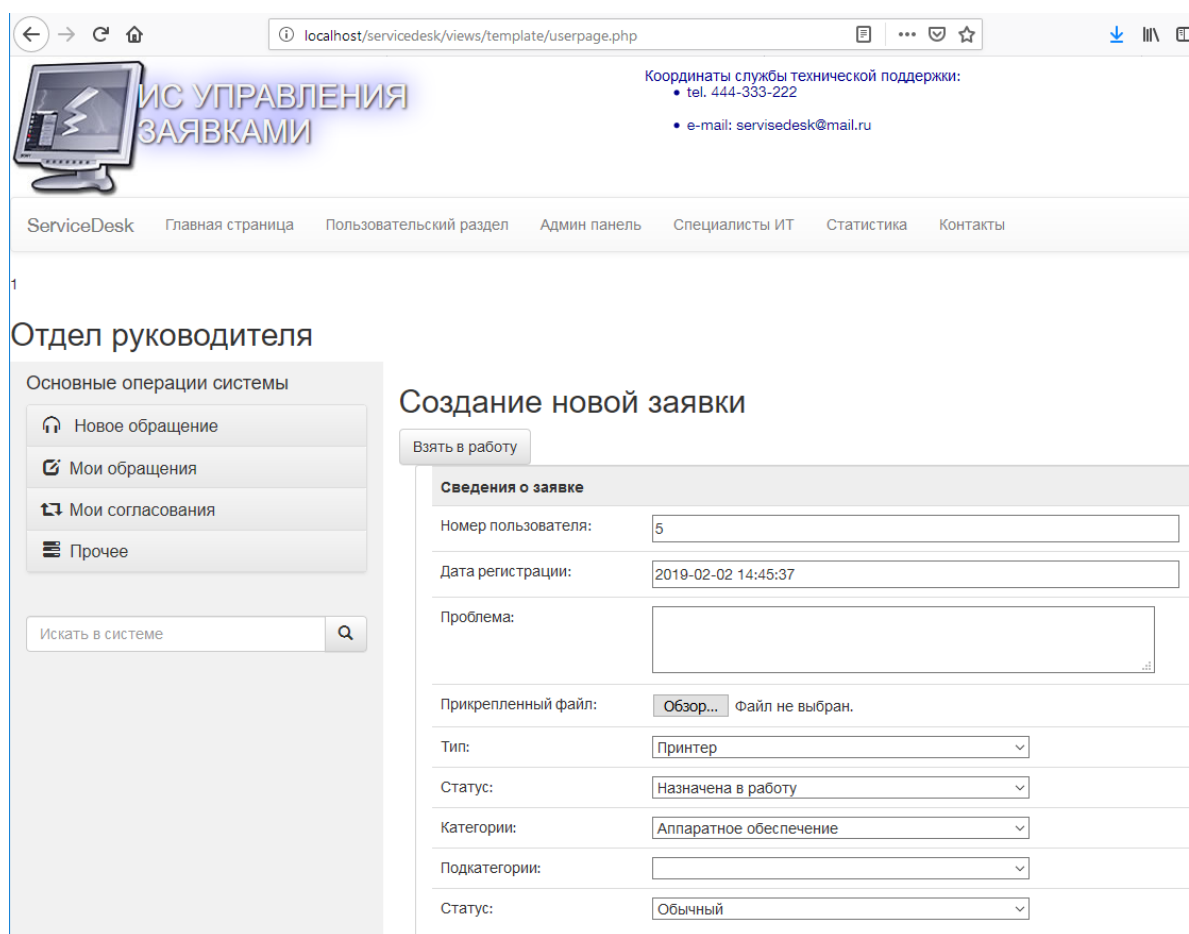


Рисунок 2.20 - Создание заявки

К доступным функциям добавлены отчеты, которые располагаются справа и могут быть просмотрены. У пользователя активны все функции, связанные с заявками.

## Выводы по главе 2

На основе изученных данных были разработаны структура ServiceDesk системы, определены ее основные модули, функции системы, роли и функции пользователей системы, место службы технической поддержки в структуре предприятия.

На начальном этапе разработки программного продукта на стадии проектирования был разработан дизайн системы, ее архитектура, концепции разработки, выбран шаблон проектирования и его элементов, написано техническое задание на разработку сайта.

На основании предыдущих этапов было выполнено проектирование автоматизированной службы технической поддержки в структуре сайта и разработан модуль автоматизированной on-line службы технической поддержки предприятия.

На последних этапах разработки проведено тестирование программы.

## Глава 3 Оценка и обоснование экономической эффективности проекта

### 3.1 Расчет показателей экономической эффективности проекта

Затраты на разработку информационной системы рассчитываются по формуле:

$K = K_{\text{п}} + K_{\text{внед}}$ , где  $K$  – затраты на разработку,  $K_{\text{п}}$  – затраты на проектирование,  $K_{\text{внед}}$  – затраты на внедрение.

Затраты на проектирование включают:

- предпроектные затраты складываются из временных расходов на проведение предпроектного обследования (Исследование и описание предметной области, обзор и анализ аналогов, постановка задачи, описание архитектуры разрабатываемой системы, выбор и обоснование методов, подходов, средств, необходимых для проектирования);
- реализацию проекта (Выполнить реинжиниринг процесса управление инцидентами, определение функционального состава системы, проектирование базы данных, проектирование пользовательского интерфейса, описание прецедентов и построение диаграмм пригодности, построение диаграммы классов, реализация программного продукта на языке php и jquery, расчет потенциальной экономической эффективности).

Расчет полных затрат на разработку проектного решения в виде информационных технологий ( $K_{\text{п}}$ ) осуществляется по формуле:

$$K_{\text{п}} = C_{\text{ФОТР}} + C_{\text{ОВФ}} + C_{\text{ЭВМ}} + C_{\text{СПС}} + C_{\text{К}}$$

где  $C_{\text{ФОТР}}$  – общий фонд оплаты труда разработчиков, руб.;

$C_{\text{ОВФ}}$  – отчисления во внебюджетные фонды с заработной платы разработчиков, руб.;

$C_{\text{ЭВМ}}$  – затраты, связанные с эксплуатацией техники, руб.;

$C_{\text{СПС}}$  – затраты на специальные программные средства, необходимые для разработки проектного решения, руб.;

$C_{\text{К}}$  – затраты на хозяйственно-операционные нужды (бумага, литература, носители информации и т.п.), руб.;

Рассчитаем фонд оплаты труда разработчиков ( $C_{\text{ФОТР}}$ ) по формуле:

$$C_{\text{ФОТР}} = \sum_{i=1}^m C_{Oi} * (1 + k_{\text{д}}) * (1 + k_{\text{у}} + k_{\text{н}}),$$

где  $C_{Oi}$  – основная заработная плата разработчика определённой категории, руб.

$t_i$  – общее время работника над проектом, в днях;

$k_{\text{д}}$  – коэффициент дополнительной зарплаты, в долях ( $k_{\text{д}}=0$ );

$k_{\text{у}}$  – районный коэффициент и северная надбавка, в долях ( $k_{\text{у}}=2,2$ )

$k_{\text{н}}$  – коэффициент, учитывающий накладные расходы организации, в долях к основной заработной плате разработчиков ( $k_{\text{н}}=0,2$ ).

$C_{Oi} = C_{\text{деньи}} + t_i$ , где

$C_{\text{деньи}}$  – среднедневная заработная плата работника  $i$ -й категории, руб./дн.;

$t_i$  - количество дней, отработанных работником  $i$ -й категории.

Затраты времени на разработку системы по каждому исполнителю принимаются, исходя из его загрузки (Таблица распределение нагрузки по проекту между разработчиками). Расчет основной заработной платы разработчиков проекта приведен в таблице из расчета, что в месяце в среднем 22 рабочий день.

Таблица 3.1 - Расчет основной заработной платы разработчиков проекта

Должность	Должностной оклад, руб.	Среднедневная ставка, руб.	Затраты времени на разработку, человеко-дней	Основная заработная плата, руб.
Руководитель	9633,74	437,90	24	10509,53
Инженер-программист	7430,86	337,77	96	34425,57
<b>Итого</b>				<b>42935,10</b>
Инженер-программист (внедрение)	7430,86	337,77	5	1688,85

$$C_{\text{ФОТР}} = \sum_{i=1}^m C_{oi} * (1 + k_d) * (1 + k_y) + k_n$$

$$= 44979,6 * (1 + 0) * (1 + 2,2) + 0,2 = 42935,10 * 3,4$$

$$= 145979,34$$

$$C_{\text{ОВФ}} = C_{\text{ФОТР}} * k_c, \text{ где}$$

$k_c$  – коэффициент, учитывающий отчисления на социальные нужды, в долях.

Таблица 3.2 - Расчет коэффициента, учитывающего отчисления на социальные нужды

№	Наименование внебюджетного фонда	Размер ставок, %
	Пенсионный фонд	22%
	Фонд социального страхования	2,9%
	Федеральный фонд обязательного медицинского страхования	5,1%
	ИТОГО	30%

$$C_{\text{ОВФ}} = C_{\text{ФОТР}} * k_c = 145979,34 * 0,3 = 43793,80$$

Посчитаем затраты, связанные с использованием вычислительной и оргтехники по формуле:

$$C_{\text{ЭВМ}} = C_{\text{ЭВМР}} + C_{\text{ЭВМИ}}$$

$C_{\text{ЭВМР}}$  – затраты на работу ЭВМ руководителя, в рублях.

$C_{\text{ЭВМИ}}$  – затраты на работу ЭВМ инженера-программиста, в рублях.

$$C_{\text{ЭВМР}} = t_{\text{ЭВМ}} * P_{\text{ЭВМч}} * K_{\text{ЭВМ}}, \text{ где}$$

$K_{\text{ЭВМ}}$  – коэффициент готовности ЭВМ, в долях.  $k_{\text{ЭВМ}} = 0,95$

$t_{\text{ЭВМ}}$  – машинное время компьютера, необходимое для разработки программного продукта, в часах.  $t_{\text{ЭВМ}} = 0,7 * 96 * 8 = 538$  часов.

$P_{\text{ЭВМч}}$  – стоимость 1 часа машинного времени,  $P_{\text{ЭВМч}} = 7,26$  руб.

$$P_{\text{ЭВМч}} = C_{\text{ЭВМч}} + A + C_{\text{рем}} / T_{\text{ЭВМ}}, \text{ где}$$

$C_{\text{ЭВМч}}$  – стоимость потребляемой в год электроэнергии, руб.;

$A$  - амортизация в год, руб.;

$C_{\text{рем}}$  - затраты на ремонт в год, руб.;

$T_{\text{ЭВМ}}$  - действительный фонд времени работы вычислительной техники, ч.

$$C_{\text{ЭВМч}} = p * T_{\text{ном}} * C_{\text{эл.энерг}}$$

$p$  – мощность, потребляемая из сети одной ЭВМ, кВт;

$T_{\text{ном}}$  – номинальный фонд времени работы ЭВМ в год, ч;

$c_{\text{эл.энерг}}$  – стоимость 1 кВт·ч электрической энергии, руб. / (кВт·ч).

Мощность, потребляемая из сети одной ЭВМ,  $p=0,3$  кВт. Стоимость 1 кВт·ч электрической энергии  $c_{\text{эл.энерг}}=3,5$  руб. / (кВт·ч). При условии, что продолжительность рабочего дня равна 8 ч, а в месяце 22 рабочих дня, номинальный фонд времени работы ЭВМ равен:

$$T_{\text{ном}} = 8 * 22 * 12 = 2112 \text{ ч в год.}$$

За год отчисления на электрическую энергию составят:

$$C_{\text{ЭВМч}} = p * T_{\text{ном}} * c_{\text{эл.энерг}} = 0,3 * 2112 * 3,5 = 2217,6 \text{ рублей.}$$

Амортизация вычислительной техники считается как 25% от ее балансовой стоимости. Стоимость ЭВМ, необходимой для работы - 40000 руб.

Амортизация вычислительной техники за год составит:

$$A = 40000 * 0,25 = 10000 \text{ рублей.}$$

Затраты на ремонт в год считаются как 4% от стоимости ЭВМ и составляют:

$$C_{\text{рем}} = 40000 * 0,04 = 1600 \text{ рублей.}$$

Действительный фонд времени работы ЭВМ в год рассчитывается как:

$$T_{\text{ЭВМ}} = T_{\text{ном}} - T_{\text{проф}},$$

где  $T_{\text{ном}}$  – номинальный годовой фонд времени работы ЭВМ, ч;  $T_{\text{проф}}$  – годовые затраты времени на профилактические работы (принимаются 10% от  $T_{\text{ном}}$ ), ч. Тогда:

$$T_{\text{ЭВМ}} = T_{\text{ном}} - T_{\text{проф}} = 2112 - 2112 * 0,1 = 1900,8.$$

Стоимость машино-часа равна:

$$P_{\text{ЭВМч}} = C_{\text{ЭВМч}} + A + C_{\text{рем}} / T_{\text{ЭВМ}} = (2217,6 + 10000 + 1600) / 1900,8 = 7,26 \text{ руб.}$$

Вычислив значения всех необходимых параметров, найдем затраты на работу ЭВМ руководителя:

$$C_{\text{ЭВМР}} = t_{\text{ЭВМ}} * P_{\text{ЭВМч}} * K_{\text{ЭВМ}} = 24 * 8 * 7,26 * 0,95 = 1324,22.$$

Найдем затраты на работу ЭВМ инженера-программиста:

$$C_{ЭВМИ} = t_{ЭВМ} * P_{ЭВМЧ} * K_{ЭВМ} = 96 * 8 * 7,26 * 0,95 = 5296,90$$

$$C_{ЭВМ} = C_{ЭВМР} + C_{ЭВМИ} = 1324,22 + 5296,90 = 6621,12$$

Поскольку затрат на специальные программные средства не было, то  $C_{СПС} = 0$  рублей.

Материалы, необходимые для выполнения работы показаны в таблице 3.3.

Таблица 3.3 - Расчет затрат на материалы, необходимые для выполнения работы показаны

Материалы	Единица измерения	Требуемое количество	Цена за единицу, руб.	Сумма, руб.
Тонер для лазерного принтера	шт.	1	4000	4000
Бумага офисная	пачка	3	250	750
Итого				4750

$$C_K = 4750 \text{ рублей.}$$

Итак, затраты на проектирование программного продукта составят:

$$K_{П} = C_{ФОТР} + C_{ОВФ} + C_{ЭВМ} + C_{СПС} + C_K = 145979,34 + 43793,80 + 6621,12 + 0 + 4750 = 201144,26.$$

Расчет затрат на внедрение программного продукта

Затраты на внедрение программного продукта ( $K_{ВПР}$ ) рассчитываются по формуле:

$$K_{ВПР} = Z_M + Z_{КТС} \times (1 + k_{ТН}) \times Z_{ПО} + Z_{ФОТВ} + Z_{ОВФ} + Z_{ЭВМ} + P_{КОМ} + P_H,$$

где  $Z_M$  – затраты на приобретение материалов, руб.;

$Z_{КТС}$  – затраты на приобретение комплекса технических средств, руб.;

$Z_{ПО}$  – затраты на приобретение программного обеспечения (включают стоимость разработанного ПП, а также других существующих ПП, необходимых для функционирования системы), руб.;

$Z_{ФОТВ}$  – затраты на оплату труда работников, занятых внедрением проекта, руб.;



$Z_{ОВФ}$  – отчисления во внебюджетные фонды с заработной платы работников, занятых внедрением проекта, руб.;

$Z_{ЭВМ}$  – затраты, связанные с эксплуатацией ЭВМ при внедрении проектного решения, руб.;

$P_{ком}$  – командировочные расходы, руб.;

$P_H$  – накладные расходы, руб.;

$k_{ТУН}$  – коэффициент транспортирования, установки и наладки комплекса технических средств, определяется действующими нормативами организации, а также спецификой конкретного проекта.

Так как для внедрения программного продукта расходных материалов не требуется, то  $Z_M = 0$ . Дополнительного приобретения компьютеров или других комплексов технических средств так же не требуется, следовательно,  $Z_{КТС} = 0$ .

Затрат на приобретение программного обеспечения нет  $Z_{ПО} = 0$  руб.

Внедрением занят один инженер программист с окладом 7430,86 руб. Время внедрения – 5 дней. По формуле рассчитываем затраты на оплату труда и отчисления во внебюджетные фонды.

$$Z_{ФОТВ} = 1688,85 * 3,4 = 5742,09 \text{ руб.}$$

$$Z_{ОВФ} = 5742,09 * 0,3 = 1722,63 \text{ руб.}$$

Затраты, связанные с эксплуатацией ЭВМ при внедрении проектного решения составят:

$$Z_{ЭВМР} = t_{ЭВМ} * P_{ЭВМЧ} * K_{ЭВМ}$$

$$Z_{ЭВМ} = 5 * 8 * 7,26 * 0,95 = 275,88 \text{ руб.}$$

Командировочные расходы при внедрении программного продукта не планируются, следовательно,  $P_{ком} = 0$ .

Так как коэффициент накладных расходов по данным организации составляет  $k_{НР} = 0,2$ , то величина накладных расходов равна  $P_H = k_{НР} * Z_{ФОТВ}$   $P_H = 5742,09 * 0,2 = 1148,42$  рублей.

Суммарные затраты на внедрение составят:

$$Z_{ВПП} = 0 + 0 + 0 + 5742,09 + 1148,42 + 275,88 + 0 + 1722,63 = 8889,02 \text{ руб.}$$

В результате проведенных расчетов получаем полные затраты на разработку с учетом этапов проектирования и внедрения:

$$K = K_{\text{п}} + K_{\text{внед}} = 201144,26 + 8889,02 = 210607,48 \text{ рублей.}$$

Оценка трудоемкости и расчет эксплуатационных затрат решения задачи.

Таблица 3.4 - Показатели, необходимые для расчета трудоемкости и эксплуатационных затрат на решение задачи

Показатель	Обозначение	Единица измерения	Значение	
			До внедрения	После внедрения
Основная заработная плата сотрудников при решении задачи	Зс	Руб.	560883,24	487020,59
Трудовые затраты сотрудника	Тс	Чел.-день	12	16
Коэффициент накладных расходов	Кнр	доля	0,2	0,2
Коэффициент северной надбавки и районный коэффициент	Ку	доля	2,2	2,2
Стоимость машино-часа	Смчр	Руб.	7,26	7,26
Время работы ЭВМ для решения задачи	Тэвмз	Час	2	2,23
Коэффициент настройки системы и оборудования	Кнсио	доля	-	0,1
Среднее количество рабочих дней в месяце	Дкол	Дни	22	22
Коэффициент прочих расходов	К	доля	0,1	0,1
Время решения задачи в год	Тзадача	дни	191,67	147,34

Для обработки одной заявки при ручном варианте решения задачи затраты составили:  $C_T + C_P = 1365,49 + 273,11 = 1638,6$  рублей.

Для обработки одной заявки при автоматизированном варианте решения задачи затраты составили:  $C_T + C_P = 1088,17 + 221,13 = 1309,3$

После проведения расчетов, подводим итог:

- трудовые затраты на обработку одной заявки при ручном варианте составили 1022,72 руб. и 750,4 – при автоматизированном;

- материальные затраты при ручном варианте составили 239,35 руб., а при автоматизированном – 153,58.

Раз в квартал сотрудники составляют квартальные отчеты, раз в год годовые отчеты о работе, на решение данной задачи у сотрудников уходит по времени в среднем 4 часа при ручном варианте, предположительно, что после внедрения системы они будут тратить на составление отчетов 10 минут. Посчитаем затраты на эту задачу при «ручном» решении в таблице 3.5.

Таблица 3.5. Расчет затрат на задачу «Составить отчеты» при ручном варианте

№	Название БП	Ответственный	Размер оплаты труда (руб./час) на ставку	Время выполнения процесса, мин	Стоимость временных ресурсов бизнес-процесса, руб., Ст	Количество повторений процесса, раз	Стоимость материального ресурса, руб. $0,2*Ст*n$	Стоимость материальных ресурсов бизнес-процесса, руб.,	Стоимость этапа процесса
1.	Составить отчеты	Системный администратор	337,77	240	1351,08	1	$0,2*1351,08*1 = 270,22$	314,01	33,78
		Руководитель	437,90	30	218,95	1	$0,2*218,95*1 = 43,79$		
2.				270	1570,03	1	314,01	314,01	1884,04

решения

В целом в год на составление отчетов тратиться:  $4*1884,04 = 7536,16$  при ручном варианте решения задачи.

Рассчитаем временные затраты на решение задачи одним сотрудником в год. В целом в год на составление отчетов тратиться:  $4*1884,04 = 7536,16$  рублей при машинном варианте решения задачи.

При ручном варианте решения задачи в день у сотрудника может уходить  $172/60=2,87$  часа на выполнение одной заявки. В день сотрудник в худшем случае может выполнить 2 заявки, затратив при этом  $2*2,87=5,74$  ч. Значит в год на решение задачи он будет тратить:  $5,74*22*12=1515,36$  часов. При этом также в решение задачи должно входить составление годовых и квартальных

отчетов. Подсчитаем сколько времени уйдет на составление отчетов в год:  
 $270/60*4=18$  часов в год. В сумме на решение задачи при ручном вводе будет  
 затрачено  $1515,36+18=1533,36$  часов в год или 191,67 чел.-дней.

Таблица 3.6. Основная заработная плата сотрудников, решающих задачу до внедрения

Должность	Должностной оклад, руб.	Среднедневная ставка, руб.	Затраты времени на решение задачи, человеко-дней	Основная заработная плата одной штатной единицы, руб	Основная заработная плата всех штатных единиц определенной категории, руб
Руководитель	9633,74	437,90	191,67	83932,29	83932,29
Системный администратор, специалист технической поддержки (2 ед.), специалист по телефонии	7430,86	337,77	191,67	64740,38	258961,52
Старший системный администратор, инженер-программист (2 ед.)	8340,32	379,11	191,67	72663,14	217989,43
<b>Итого</b>					<b>560883,24</b>

Таблица 3.7. Основная заработная плата сотрудников, решающих задачу после внедрения

Должность	Должностной оклад, руб.	Среднедневная ставка, руб.	Затраты времени на решение задачи, человеко-дней	Основная заработная плата одной штатной единицы, руб	Основная заработная плата всех штатных единиц определенной категории, руб
Руководитель	9633,74	437,90	147,34	64520,19	64520,19
Системный администратор, специалист технической поддержки,	7430,86	337,77	147,34	49767,03	199068,13

специалист по телефонии					
Старший системный администратор	8340,32	379,11	147,34	55858,07	223432,27
<b>Итого</b>					<b>487020,59</b>

При машинном варианте решения задачи в день у сотрудника может уходить  $134/60=2,23$  часа на выполнение одной заявки. Допустим, что в день сотрудник выполнит, так же, как и при ручном варианте, 2 заявки, затратив при этом  $2*2,23=4,46$  ч. Значит в год на решение задачи он будет тратить:  $4,46*22*12=1177,44$  часов. При этом также в решение задачи должно входить составление годовых и квартальных отчетов. Подсчитаем сколько времени уйдет на составление отчетов в год:  $20/60*4=1,3$  часа в год. В сумме на решение задачи при машинном варианте будет затрачено  $1177,44+1,3=1178,74$  часов в год или 147,34 чел.-дней.

Расчет эксплуатационных затрат при ручном варианте решения задачи:

$K_{эр} = C_{фотрруч} + C_{овф} + C_{эвм} + C_{нак} + C_a$ , где

$K_{эр}$  – годовые эксплуатационные затраты при ручной обработке, руб;

$C_{фотрруч}$  – затраты на оплату труда сотрудников при ручном варианте решения задачи, руб;

$C_{овфроч}$  – отчисления во внебюджетные фонды с заработной платы сотрудников, руб.;

$C_{эвм}$  - затраты, связанные с эксплуатацией техники, руб.;

$C_{нак}$  – накладные расходы, руб;

$C_{нак} = 0,2 * C_{фотрруч} = 1907003,02 * 0,2 = 381400,60$

$C_{фотрруч} = m \sum_{i=1}^n C_{oi} * ((1+K_d) * (1+K_y) + K_{нр}) = 560883,24 * ((1+0) * (1+2,2)+0,2) = 1907003,02$  рублей.

$C_{овфроч} = C_{фотрруч} * K_c = 1907003,02 * 0,3 = 572100,91$  рублей.

$C_{эвм} = 0,7 * 1533,36 * 7,26 * 0,95 = 1073,35 * 7,26 * 0,95 = 7402,91$

Учитывая то, что в решении задачи участвуют 10 сотрудников, то:

$C_{эвм} = 7402,91 * 10 = 74029,1$

$K_{\text{эр}} = C_{\text{фотрр}} + C_{\text{овф}} + C_{\text{ЭВМ}} + C_{\text{нак}} = 1907003,02 + 572100,91 + 74029,1 + 381400,60 = 2\,934\,533,63$  рублей.

$K_{\text{эм}} = C_{\text{фотм}} + C_{\text{овфм}} + C_{\text{ЭВММ}} + C_{\text{накм}}$

$K_{\text{эм}}$  – годовые эксплуатационные затраты при машинной обработке, руб;

$C_{\text{фотм}}$  – затраты на оплату труда сотрудников при машинном варианте решения задачи, руб;

$C_{\text{овфм}}$  – отчисления во внебюджетные фонды с заработной платы, руб.;

$C_{\text{ЭВММ}}$  - затраты, связанные с эксплуатацией техники, руб.;

$C_{\text{накм}}$  – накладные расходы, руб;

$C_{\text{нак}} = 0,2 * C_{\text{фотм}} = 1655870,01 * 0,2 = 331174,00$

$C_{\text{фотм}} = m \sum_{i=1}^n C_{oi} * ((1+K_{д}) * (1+K_{у}) + K_{нр}) = 487020,59 * ((1+0) * (1+2,2) + 0,2) = 1655870,01$  рублей.

$C_{\text{овфм}} = C_{\text{фотм}} * K_{с} = 1655870,01 * 0,3 = 496761,00$  рублей.

$C_{\text{ЭВММ}} = 0,7 * 1178,72 * 7,26 * 0,95 = 825,10 * 7,26 * 0,95 = 5690,74$

Учитывая то, что в решении задачи участвуют 10 сотрудников и сервер, то:

$C_{\text{ЭВМ}} = 7402,91 * 11 = 62598,17$

$K_{\text{эм}} = C_{\text{фотм}} + C_{\text{овфм}} + C_{\text{ЭВММ}} + C_{\text{накм}} = 1655870,01 + 496761,00 + 62598,17 + 331174,00 = 2\,546\,403,18$  рублей.

Таблица 3.8. Сравнительная таблица эксплуатационных затрат машинного и ручного вариантов решения задачи

Наименование затрат	Ручной вариант решения задачи, руб.	Машинный вариант решения задачи, руб.
Затраты на оплату труда сотрудников	1907003,02	1655870,01
Отчисления во внебюджетные фонды	572100,91	496761,00
Затраты, связанные с эксплуатацией техники	74029,1	62598,17
Накладные расходы	381400,60	331174,00
ИТОГО	2934533,63	2546403,18

Таблица 3.9. Сравнительная таблица показателей машинного и ручного вариантов решения задачи

Наименование характеристик	Показатели	Ручной вариант решения задачи	Машинный вариант решения задачи

Наименование характеристик	Показатели	Ручной вариант решения задачи	Машинный вариант решения задачи
Производительность	Количество учтенных заявок на сотрудника в день, шт.	2	3
	Количество обслуженных пользователей на сотрудника в день, шт.	2	3
Длительность	Время обработки одной заявки, мин	172	134
Стоимость	Стоимость обработки одной заявки	1638,6	920,87
Результативность	Количество обработанных заявок	2	3
Эффективность	Фактическое количество учтенных заявок/затраты	1/1638,6	1/920,87

В данном параграфе проведены расчеты основных показателей, связанных с финансовыми потоками при разработке и использовании информационной системы.

### **Выводы по главе 3**

На основе произведенных расчётов и определенной оценке экономической эффективности информационной системы, можно сделать выводы по потенциальным результатам, которые можно получить от внедрения информационной системы:

- внедрение проекта информационной системы по учету и обработки заявки (Service Desk) в службе технической поддержки ООО «ЛенАвтоТехника» позволит сократить временные затраты на решение задачи на 354 часа в год, что приведет к сокращению годовых эксплуатационных затрат с 2934533,63 руб. до 2546403,18 руб;

- вычисленный коэффициент экономической эффективности выше нормативного коэффициента эффективности капитальных затрат 1,85, что говорит об эффективности проекта;

- срок окупаемости капитальных вложений составляет 6,47 месяца;

- внедрение проекта позволит повысить показатели бизнес-процесса управление инцидентами и устранить выявленные в аналитической части недостатки бизнес-процесса «Как есть».

Опираясь на вышеперечисленные результаты сделаем вывод о том, что разработка, внедрение и эксплуатация спроектированной системы экономически и технически обосновано и целесообразно.



## **ЗАКЛЮЧЕНИЕ**

В рамках выпускной квалифицированной работы была разработана информационная система учета и обработки заявок (Service Desk), поступающих от пользователей для службы технической поддержки автомобильной сервисной компании ООО «ЛенАвтоТехника», путем решения поставленных задач по обслуживанию компьютерной техники задействованной в сфере управления и, в небольшой мере, в процессе проектирования.

Дана общая характеристика предприятия и его деятельности, приведена его организационная структура, выполнен анализ существующего уровня автоматизации, проанализирован подлежащий автоматизации процесс по обработке заявок пользователей, определены требования к системе по видам обеспечения.

Выполнена постановка задачи для проектируемой системы, определены назначение и цель создания системы, описан перечень объектов в интересах, которых решаются задачи, периодичность решения задачи, входная и выходная информация и формализация задачи.

В ходе проектной части работы были описаны архитектура системы, ее серверная и клиентская части, осуществлен выбор и обоснование методов, подходов и средств проектирования, построены функциональная модель учета заявок, диаграмма вариантов использования, диаграмма потоков данных, концептуальная, логическая и физическая модели базы данных, спроектирован прототип пользовательского интерфейса.

В экономической части работы решена задача по оценки эффективности проекта.

Система была спроектирована для автоматизации процесса управления инцидентами и повышения эффективности работы специалистов службы технической поддержки предприятия ООО «ЛенАвтоТехника». Особенности спроектированной информационной системы выступают легкость и простота в использовании и не требующая специального обучения специалистов компании.

## СПИСОК ИСПОЛЬЗУЕМОЙ ЛИТЕРАТУРЫ

### *Научная и методическая литература*

1. Агальцов, В.П. Базы данных. В 2-х т.Т. 1. Локальные базы данных: Учебник / В.П. Агальцов. - М.: ИД ФОРУМ, НИЦ ИНФРА-М, 2013. – 352 с.
2. Буч Г., Рамбо Д., Джекобсон А. Язык UML. Руководство пользователя. - М.: ДМК Пресс, 2001.
3. Вдовенко, Л.А. Информационная система предприятия: учеб. пособие / Л.А. Вдовенко. – М.:Инфра-М, 2011. – 240 с.
4. Вендров, А.М. Проектирование программного обеспечения экономических информационных систем: Учеб. - М.: Финансы и статистика, 2000.
5. Вендров, А.М. CASE-технологии. Современные методы и средства проектирования информационных систем. М. : Финансы и статистика, 1998. 176 с.
6. Грекул В.И., Денищенко Г.Н., Коровкина Н.Л. Проектирование информационных систем. Интернет-университет информационных технологий. / В.И. Грекул, Г.Н. Денищенко, Н.Л. Коровкина // ИНТУИТ.ру. – 2008.
7. Исаев, Г.Н. Информационные системы в экономике / Г.Н.Исаев. – М.: Омега-Л, 2008.- 462 с.
8. Данилин А. Архитектура и стратегия. "Инь" и "янь" информационных технологий / А.Данилин, А. Слюсаренко. – М.: Интернет-ун-т Информ. Технологий, 2011. – 504 с.
9. Кватрани, Т. Rational Rose 2000 и UML. Визуальное моделирование. - М.: ДМК Пресс, 2001.
10. Коваленко, В.В. Проектирование информационных систем: учебное пособие / В.В. Коваленко. – М.: Форум, 2012. – 320 с.
11. Ларман, К. Применение UML и шаблонов проектирования. - М.: Издательский дом «Вильямс», 2001.
12. Леоненков, А.В. Самоучитель UML. - СПб.: БХВ-Петербург, 2001.

13. Мандел, Т. Разработка пользовательского интерфейса. - М: ДМК Пресс, 2001.
14. Мартин, Дж. Организация баз данных в вычислительных системах. - М.: Мир, 1980.
15. Чен, П. Модель «сущность-связь» - шаг к единому представлению данных // СУБД. 1995. №3. С. 137-158.
16. Петров, В.И. Информационные системы. СПб. : Питер, 2002. 688 с.
17. Скотт Беркун, Искусство управления IT -проектами / Беркун Скотт. – СПб.: Питер, 2010. – 432 с.
18. Солдатов, В.П. Управление программными проектами / В.П. Солдатов. – М.: Бином, 2010. – 382 с.
19. Томсон, Л. Розробка Web -приложений на PHP і MySQL: Пер. з англ./Лаура Томсон, Люк Веллинг. - 2-е видавництво, испр. - СПб: ТОВ ДиаСофтЮП, 2003. - 672 с.
20. Федоров, Н.В. Проектирование информационных систем на основе современных CASE-технологий. – М.: МГИУ, 2008. – 287 с.
21. Черемных С.В., Ручкин В.С., Семенов И.О. Структурный анализ систем IDEF-технологии. / С.В. Черемных, В.С. Ручкин, И.О. Семенов – М.: Финансы и статистика, 2001.

*Литература на иностранном языке*

22. Sklar, D. Learning PHP and MySQL. - Boston: O'Reilly, 2016. – 395 с.
23. Axelos, A. ITIL Practitioner Guidance. Stationery, 2017. – 231 с.
24. Peter, F. ITIL For Dummies. - West Sussex: Stationery, 2012. – 357 с.
25. Dubois, P. MySQL Cookbook. - Sebastopol: O'Reilly, 2014. – 354 с.
26. Nield, T. Getting Started with SQL. - Sebastopol: O'Reilly, 2016. – 450 с.