# Введение

В рамках дипломной работы была поставлена задача разработать HelpDesk - автоматизированную информационную систему для автоматизации работы ИТ отдела предприятия с пользователями компьютерной и оргтехники при возникновении у них проблем решение которых лежит в сфере деятельности отдела.

Крупные организации располагают на сегодня сложной ИТ-инфраструктурой, непрерывное функционирование всех элементов которой на должном уровне является обязательным условием для выполнения организацией своих основных функций. Поддержка этой инфраструктуры в рабочем состоянии является одной из основных функций ИТ-службы предприятия. Необходимость решения этих вопросов очевидна: неспособность оперативно и качественно обрабатывать запросы пользователей, минимизировать негативные последствия сбоев - явные признаки слабой организации ИТ-службы и помеха для работы всей компании.

Решение этих вопросов требует комплекса организационно -технических мер

Системы HelpDesk позволят ИТ-службам обеспечить качественное выполнение этих функции.

Service desk состоит из следующих логических компонентов:

* модуль регистрации заявок об инцидентах
* база данных заявок
* система отслеживания статуса заявки и оповещения
* база знаний
* панель администрирования
* модуль отчетности

# 1 Общая часть.

## 1.1 Характеристика предприятия.

Датой рождения открытого акционерного общества «Салаватнефтехимремстрой» принято считать 1 января 1970 года.

В конце 1969 года Министерство нефтеперерабатывающей и нефтехимической промышленности СССР приняло решение о создании в городе Салавате на базе южной группы строительных, ремонтно-строительных управлений и участков, входящих в трест «Башнефтехимремстрой» (трест «БНХРС»), самостоятельного предприятия треста «Салаватнефтехимремстрой».

В 1995 году в связи с акционированием трест «Салаватнефтехимрестрой» был преобразован в открытое акционерное Общество «Салаватнефтехимремстрой».

Виды выполняемых работ:

Общестроительные работы по возведению и ремонту промышленных зданий и сооружений, выполнение геодезических, подготовительных, свайных и специальных бетонных работ

Огнеупорные работы на высокотемпературных тепловых агрегатах

Изготовление узлов, ремонт оборудования с применением термообработки на промышленных предприятиях нефтехимии, нефтепереработки, включая объекты, подведомственные Госгортехнадзору России

Изготовление элементов, монтаж и ремонт тепловой изоляции

Монтаж и капитальный ремонт технологического оборудования и трубопроводов, работающих под давлением, трубопроводов пара и горячей воды; градирен водооборотных узлов промышленных предприятий; промышленных систем вентиляции; бетонных, железобетонных и стальных конструкций

Ремонт, прокладка подземных коммуникаций промышленных предприятий и тепловых сетей

Кровельные, отделочные и сантехнические работы

Ремонт и строительство автомобильных дорог, работы по благоустройству территории

Контроль оборудования и материалов разрушающими и неразрушающими методами на базе собственной испытательной лаборатории

Услуги на базе проектных офисов: лицензирование, базовое и рабочее проектирование, экспертиза ПСД, поставка оборудования, СМР, ПИР, ввод в эксплуатацию.

## 1.2 Обоснование актуальности и постановка задачи

Внедрение службы Service Desk целесообразно по многим причинам. Многие современные средства автоматизации предусматривают возможность регистрации заявок самим пользователем (через Web-формы, email письма). Однако пользователь порой не в состоянии квалифицированно определить суть проблемы. Но даже если он может провести самостоятельную классификацию, все равно требуется уточнение (или, как минимум, подтверждение) специалиста. Автоматизированный ввод инцидентов самим пользователем не отменяет механизма ввода и модерации специалистом службы Service Desk, поскольку в ряде случаев пользователь просто не имеет возможности завести заявку (инцидент) для автоматической обработки.

Service Desk обеспечивает единую точку входа. Это очень важно, поскольку отпадает необходимость выстраивать персональные отношения с ИТ-специалистами, и пользователь всегда знает, куда ему обратиться с проблемами.

При отказе от Service Desk весьма вероятно возникновение «очередей» заявок у высококвалифицированных ИТ-специалистов, которые будут совершенно неэффективно тратить свои ресурсы. Запуск же квалифицированной службы Service Desk позволяет спустя некоторое время разрешать до 70% заявок на первой линии благодаря накоплению базы знаний и другим преимуществам правильно организованных процессов. Это экономит время высокооплачиваемых специалистов, и они могут направить его для решения более важных задач. Оставшиеся 30% заявок маршрутизируются таким образом, что гарантируется их скорейшее решение.

Три главных принципа в организации ServiceDesk:

Правила внедрения:

* полное понимание зоны ответственности ИТ-подразделения
* кто из пользователей может подать заявку, и по каким услугам
* сроки и приоритеты выполнения заявок, сгенерированные пользователями различных подразделений

Управление ожиданиями пользователей:

* при любом обращении пользователя формализуется уведомление (информирование) о возможностях, сроках решения заявки и ответственном за заявку

Правило единого окна:

* все обращения – через единственную точку входа в ИТ- подразделение (ответственного дежурного, специалиста первой линии поддержки)

Разобравшись с необходимостью внедрения, определившись с правилами, по которым будет функционировать будущая система, что получим в итоге:

Результат для ИТ руководителей:

* точные данные о качестве работы ИТ-подразделения
* четкую картину о том, что происходит в ИТ-подразделении

Результат для ИТ специалистов:

* полную картину о состоянии ИТ инфраструктуры
* полную информацию о ПО

И самое главное – предприятие получит информационную систему на мощной, гибкой и функциональной платформе, которая позволяет организовывать управление заявками, проблемами, изменениями, что способствует поступательному развитию предприятия.

Предприятие, внедряющая данную АИС, является крупнейшей строительной организацией по сооружению нефтехимических объектов в южном регионе республики Башкортостан. Парк компьютерной техники насчитывает более 1800 компьютеров, 900 единиц оргтехники, 80 единиц серверного оборудования.

Среднее количество заявок ИТ службу предприятия:

* 30 по проблемам, связанным ПК и программным обеспечением
* 10 по заправке и ремонту оргтехники
* 10 по функционированию автоматизированных систем предприятия

Отел состоит из трех секторов, решающих разные типы задачи.

Рисунок 1.1 – Структура ИТ отдела предприятия

Сектор информационных технологий и связи

Решает все проблемы, связанные с компьютерной и оргтехникой, функционированием автоматизированных систем предприятия, сетевой инфраструктуры.

Сектор связи.

Поддерживает работоспособность системы связи и оповещения

Сектор информационных ресурсов и аналитических систем

Вопросы планирования, закупки оборудования и программного обеспечения.

Для АИС ставятся задача централизации приема заявок от пользователей, распределение по секторам и по специалистам, исходя из характера проблемы.

В АИС выделяется несколько автоматизированных рабочих места:

1. АРМ «Диспетчера»

Диспетчер отдела должна иметь возможность заводить заявку, отслеживать ее состояние, выводить на печать статистику по различным параметрам

1. АРМ «Инженер»

Инженер или специалист отдела должен видеть свои заявки, их состояние, оставлять комментарий, менять ее состояние, получать быструю статистику по заявкам

1. АРМ «Пользователь»

Пользователь открыв по ссылке веб-ресурс должен получить возможность оставить заявку в ИТ отдел со следующими параметрами - подразделение, ФИО, телефон, IP адрес, описание проблемы. Получить номер заявки для связи с диспетчером по поводу решения проблемы.

1. АРМ «Администратор»

Рабочее место администратора системы, для поддержания АИС в актуальном состоянии (списки подразделений, проблем, пользователей и т.п.)

## 1.3 Обзор существующих решений автоматизации задачи.

### 1.3.1 АИС «Fast HelpDesk»

Большинство компаний, активно использующие информационные технологии, в процессе роста сталкиваются с необходимостью пересмотра существующей практики эксплуатации информационных систем, в первую очередь решения таких связанных вопросов как эффективной организации поддержки пользователей (help desk) и устранения влияющих на работу пользователей сбоев.

Необходимость решения этих вопросов очевидна: неспособность оперативно и качественно обрабатывать запросы пользователей, минимизировать негативные последствия сбоев - явные признаки слабой организации ИТ-службы и помеха для работы всей компании.

Решение этих вопросов требует комплекса организационно-технических мер:

* создание для пользователей удобного и дружественного «интерфейса» к ИТ-службе для получения необходимой помощи и поддержки. Большинство организаций создают для этого отдельную группу или службу поддержки пользователей, help desk;
* организация правильного взаимодействия специалистов внутри ИТ-службы,  позволяющая быстро привлечь специалистов с нужной компетенцией и скоординировать работу различных специалистов, обеспечить своевременность выполнения работ;
* автоматизация деятельности для повышения эффективности работы, обеспечения оперативного контроля над ситуацией, снижения ошибок в работе и накопления знаний.

Часто стремительный рост требований и ожиданий бизнеса требует решительного внедрения перечисленных мер. Многие компании адекватно оценивают сложности, связанные с постановкой задачи, риски, связанные с требуемыми организационными изменениями, выбором и внедрением подходящего средства автоматизации, а также преимущества использования проверенного опыта. Однако ограничения бюджета могут быть «камнем преткновения» в привлечении внешней помощи.

Услуга является результатом обобщения многолетнего опыта IT Expert в области совершенствования работ ИТ-служб с использованием лучших международных практик (ITIL®) и нацелена на решение следующих задач:

* организация взаимодействия с пользователями ИТ-услуг;
* создание каталога ИТ-услуг, как важного инструмента фиксации обязательств и норм по обслуживанию пользователей;   
  построение управляемой деятельности по устранению сбоев в ИТ-инфраструктуре;
* организация обработки стандартных запросов на обслуживание;
* устранение причин сбоев в ИТ-инфраструктуре;
* организация выполнения и контроля регламентных работ по обслуживанию ИТ-инфраструктуры.

Отличительная черта услуги "Fast HelpDesk" - высокая степень предварительной проработки:

* заранее сформированный пакет документации с описание регламентов, инструкций, необходимых шаблонов, подготовленный к «тонкой настройке» в ходе работ для учета специфики организации;
* предварительно настроенный прототип системы автоматизации процессной деятельности на базе гибкого и масштабируемого продукта OmniTracker, готового к дальнейшему росту организации;
* описанная технология проведения работ.

Такая степень проработки обеспечивает серьезные преимущества:

* сокращение сроков получения результатов – продолжительность работ немногим больше двух месяцев;
* сокращение стоимости работ;
* снижение рисков за счет применения неоднократно проверенного сбалансированного решения.
* Высокая степень проработки – результат анализа множества выполненных проектов. Уже определены успешные варианты решений, происходит только выбор из их исчерпывающего перечня с учетом специфики конкретной организации.
* Услуга является результатом обобщения многолетнего опыта IT Expert в области совершенствования работ ИТ-служб с использованием лучших международных практик (ITIL®) и нацелена на решение следующих задач:
* организация взаимодействия с пользователями ИТ-услуг;
* создание каталога ИТ-услуг, как важного инструмента фиксации обязательств и норм по обслуживанию пользователей;   
  построение управляемой деятельности по устранению сбоев в ИТ-инфраструктуре;
* организация обработки стандартных запросов на обслуживание;
* устранение причин сбоев в ИТ-инфраструктуре;
* организация выполнения и контроля регламентных работ по обслуживанию ИТ-инфраструктуры.
* Отличительная черта услуги "Fast HelpDesk" - высокая степень предварительной проработки:
* заранее сформированный пакет документации с описание регламентов, инструкций, необходимых шаблонов, подготовленный к «тонкой настройке» в ходе работ для учета специфики организации;
* предварительно настроенный прототип системы автоматизации процессной деятельности на базе гибкого и масштабируемого продукта OmniTracker, готового к дальнейшему росту организации;
* описанная технология проведения работ.

### 1.3.1 АИС «Gestionnaire libre de parc informatique» (GLPI).

GLPI - это веб-приложение, разработанное на PHP для работы с ним необходим только доступ к серверу и браузер.

Из имевшегося функционала рассмотрим

Оборудование — сгруппированы по следующим категориям:

* «компьютеры»;
* «мониторы»;
* «программное обеспечение»;
* «сети»;
* «устройства» (другое оборудование);
* «принтеры»;
* «картриджи»;
* «расходные материалы»;
* «телефоны».

Возможность создания шаблона, расширенные подключения устройств. К одному компьютеру возможно подключить несколько мониторов, принтеров, устройств, телефонов.

Выбрав в меню – «Поддержка» (Service Desk), можно принимать обращения и назначать ответственных, планировать действия.

* Меню «Управление» сосредоточены взаимоотношения с внешними поставщиками:
* «контакты»;
* «поставщики»;
* «договора»;
* «документы» (сопроводительная информация с возможностью прикрепления почти ко всем элементам GLPI)
* Меню «Управление»:
* «заметки» (с возможностью планирования);
* «база знаний» (информация представлена здесь в иерархическом виде, позволяющая настроить публичный просмотр или только определенным группам);
* «в пользовании» (отображаются временно доступные для использования элементы, программное или аппаратное обеспечение);
* «Отчеты» — говорит само за себя.
* Администрирование;
* «Пользователи» (возможен импорт из AD или другой LDAP);
* «группы»;
* «организации»;
* «правила» (Назначение прав и организации для пользователей, Правила для заявок, Правила назначения категории для ПО);
* «профили» (настройка профилей групп пользователей);
* «Данные» (создание архивной копии в Sql или XML);
* «Журналы» (подробная информация обо всех изменения в системе, возможно, настроить протоколирование)
* «Установки» более тонкая настойка компонентов системы.

Возможно, осуществлять поиск в системе по различным параметрам.

Внесение параметров происходит довольно просто и быстро, достаточно первоначально настроить шаблоны для всех основных CI.

При получении заявки вносим логин (или имя, система автоматически предложит список с наиболее точными вариантами) заявителя (изначально у нас было принято получить логин пользователя, GLPI работает по тому же принципу), затем высвечивается весь список устройств и программ, установленных или сопряженных с тем компьютером, на котором работает обратившийся.

Здесь и возникла проблема. Принтеры невозможно внести выше описанным способом, в списке устройств они не отображаются.

Система спроектирована довольно продуманно, поэтому даже при использовании такого способа, внесение информации происходит довольно быстро.

При заполнении «Заявки» возможно, внести следующую информацию:

* Заказчика (заявителя);
* группу, к которой относится обратившийся;
* устройство или программу;
* получено через: телефон, электронную почту, лично, письменно;
* приоритет;
* категорию;
* назначить технического специалиста;
* статус;
* у страняя инцидент, специалисту дана возможность:
* добавить описание проведенных работ, а также прикрепить файл;
* указать денежные затраты на решение, материалы;
* присутствует возможность запланировать действия и назначить исполнителя;
* внести общее время на устранение инцидента.

На сайте разработчика существует довольно большое количество плагинов, расширяющих работу системы.

## 1.4 Выбор аппаратного и программного обеспечения.

Разрабатываемая система является клиент-серверной системой, то есть ядро ИС устанавливается на серверном оборудовании, а пользователи получает к нему доступ путем подключения через клиентское программное обеспечение.

Серверная часть обеспечивает хранение, обработку информации, а клиентская ввод-вывод данных и отображение информации пользователю.

Так как ИС рассчитана на внутренние потребности, то был выбран вариант, когда серверная часть будет находиться на территории предприятия, то есть установку, настройку и поддержку сервера будет обеспечивать отдел ИТ.

Данный вариант потребует финансовых вложений на покупку серверного оборудования, но окупится благодаря тому, что размещение ресурса и поддержка сервера осуществляется собственными силами.

В качестве сервера используется HP ProLiant DL160 Gen8

Таблица 1.1 – Конфигурация сервера

|  |  |
| --- | --- |
| Процессор | Intel® Xeon® E5-2640 (6-ядерный, 2,5 ГГц, 15 МБ, 95 Вт) |
| Оперативная память | 16 ГБ (4x4 ГБ) RDIMM |
| HDD диски | 4 шт., жесткий диск HP 3TB 6G SATA 7.2K rpm LFF (3.5-inch) SC Midline 1yr Warranty Hard Drive |
| Операционная система | Ubuntu 12.04 LTS |

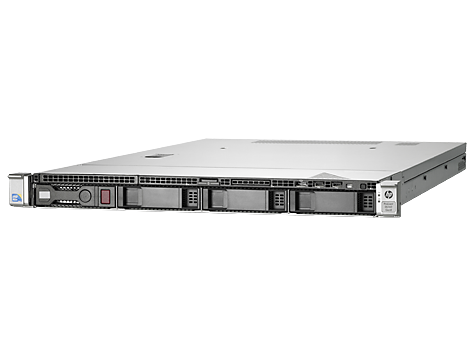


Рисунок 2.3 – Внешний вид сервера для АИС

На предприятии разработаны стандарт рабочего места в котором установлена стандартная конфигурация компьютера для пользователя.

В настоящий момент стандартом рабочего места является конфигурация компьютера HP Pro 3500 в корпусе Microtower

Таблица 2.2 – Конфигурация компьютера

|  |  |
| --- | --- |
| Процессор | Intel® Core™ i3-3220 с графической картой Intel HD 2500 (3,3 ГГц, 3 МБ кэш-памяти, 2 ядра) |
| Оперативная память | 8 Гб 1600 МГц DDR3 SDRAM |
| HDD диски | 500 Гб SATA (7200 об./мин) |
| Операционная система | Windows 8 Pro 64 |
| Монитор | HP Pavilion 23xi, 23 дюйма, технология IPS, LED-подсветка, FullHD |

Программное обеспечение сервера.

Операционная система.

Ubuntu 14.04 LTS 64-bit server

Операционная система, основанная на Debian GNU/Linux. Основным разработчиком и спонсором является компания Canonical. В настоящее время проект активно развивается и поддерживается свободным сообществом.

По утверждениям Canonical, операционную систему Ubuntu использует примерно 20 миллионов пользователей, что делает его самым популярным дистрибутивом Linux для десктопов. Он является 4-м в списке самых популярных дистрибутивов Linux для веб-серверов и его популярность быстро растёт.

Обычно новые версии дистрибутива выходят каждые полгода и поддерживаются обновлениями безопасности в течение полутора лет, а версии LTS, выходящие раз в 2 года, поддерживаются в течение 5 лет, как серверные, так и десктопные варианты. Начиная с версии 12.04 поддержка LTS-версий производится в течение 5 лет независимо от версии, в отличие от предыдущих 3 лет для настольных версий и 5 лет для серверных.

Веб-сервер

Apache

Самый популярный веб-сервер в Интернете. Несколько лет разработки сделали его чрезвычайно удобным, быстрым, лёгким в настройке и надёжным. Apache является кроссплатформенным ПО, поддерживает операционные системы Linux, BSD, Mac OS, Microsoft Windows, Novell NetWare, BeOS.

Основными достоинствами Apache считаются надёжность и гибкость конфигурации. Он позволяет подключать внешние модули для предоставления данных, использовать СУБД для аутентификации пользователей, модифицировать сообщения об ошибках и т. д. Поддерживает IPv6.

Сервер был написан в начале 1995 года и считается, что его имя восходит к шуточному названию «a patchy», так как он устранял ошибки популярного тогда сервера Всемирной паутины NCSA HTTPd 1.3. В дальнейшем, с версии 2.х, сервер был переписан заново и теперь не содержит кода NCSA. На данный момент разработка ведётся в ветке 2.4, а в версиях 1.3 и 2.0 производятся лишь исправления ошибок безопасности. На текущий момент последняя версия ветки 2.4 — 2.4.9 ([17 марта](http://ru.wikipedia.org/wiki/17_%D0%BC%D0%B0%D1%80%D1%82%D0%B0) [2014](http://ru.wikipedia.org/wiki/2014)), 2.2 — 2.2.27 ([26 марта](http://ru.wikipedia.org/wiki/26_%D0%BC%D0%B0%D1%80%D1%82%D0%B0) 2014), для первой версии это 1.3.42.

Веб-сервер Apache разрабатывается и поддерживается открытым сообществом разработчиков под эгидой Apache Software Foundation и включён во многие программные продукты, среди которых СУБД Oracle и IBM WebSphere.

С апреля 1996 и до настоящего времени является самым популярным HTTP-сервером в Интернете. Статистика Netcraft показывает следующие данные об использовании Apache:

в августе 2007 года он работал на 51 % всех веб-серверов

в мае 2009 года — на 46 %

в январе 2011 года — на 59 %, т.е. более чем на 160 млн сайтов.

Система конфигурации

Система конфигурации Apache основана на текстовых конфигурационных файлах. Имеет три условных уровня конфигурации:

Конфигурация сервера (httpd.conf).

Конфигурация виртуального хоста (httpd.conf c версии 2.2, extra/httpd-vhosts.conf).

Конфигурация уровня директории (.htaccess).

Имеет собственный язык конфигурационных файлов, основанный на блоках директив. Практически все параметры ядра могут быть изменены через конфигурационные файлы, вплоть до управления MPM. Большая часть модулей имеет собственные параметры.

Часть модулей использует в своей работе конфигурационные файлы операционной системы (например /etc/passwd и /etc/hosts).

Помимо этого, параметры могут быть заданы через ключи командной строки.

База данных

MySQL

Свободная реляционная система управления базами данных. Разработку и поддержку MySQL осуществляет корпорация Oracle, получившая права на торговую марку вместе с поглощённой Sun Microsystems, которая ранее приобрела шведскую компанию MySQL AB. Продукт распространяется как под GNU General Public License, так и под собственной коммерческой лицензией. Помимо этого, разработчики создают функциональность по заказу лицензионных пользователей. Именно благодаря такому заказу почти в самых ранних версиях появился механизм репликации.

MySQL является решением для малых и средних приложений. Входит в состав серверов WAMP, AppServ, LAMP и в портативные сборки серверов Денвер, XAMPP. Обычно MySQL используется в качестве сервера, к которому обращаются локальные или удалённые клиенты, однако в дистрибутив входит библиотека внутреннего сервера, позволяющая включать MySQL в автономные программы.

Гибкость СУБД MySQL обеспечивается поддержкой большого количества типов таблиц: пользователи могут выбрать как таблицы типа MyISAM, поддерживающие полнотекстовый поиск, так и таблицы InnoDB, поддерживающие транзакции на уровне отдельных записей. Более того, СУБД MySQL поставляется со специальным типом таблиц EXAMPLE, демонстрирующим принципы создания новых типов таблиц. Благодаря открытой архитектуре и GPL-лицензированию, в СУБД MySQL постоянно появляются новые типы таблиц.

26 февраля 2008 года Sun Microsystems приобрела MySQL AB за $1 млрд, 27 января 2010 года Oracle приобрела Sun Microsystems за $7,4 млрд и включила MySQL в свою линейку СУБД.

Сообществом разработчиков MySQL созданы различные ответвления кода, такие как Drizzle (англ.), OurDelta, Percona Server, и MariaDB. Все эти ответвления уже существовали на момент поглощения компании Sun корпорацией Oracle.

Язык программирования

PHP – Скриптовый язык программирования общего назначения, интенсивно применяемый для разработки веб-приложений. В настоящее время поддерживается подавляющим большинством хостинг-провайдеров и является одним из лидеров среди языков программирования, применяющихся для создания динамических веб-сайтов.

JavaScript — прототипно-ориентированный сценарный язык программирования. Является диалектом языка ECMAScript.

JavaScript обычно используется как встраиваемый язык для программного доступа к объектам приложений. Наиболее широкое применение находит в браузерах как язык сценариев для приданияинтерактивности веб-страницам.

Основные архитектурные черты: динамическая типизация, слабая типизация, автоматическое управление памятью, прототипное программирование, функции как объекты первого класса.

На JavaScript оказали влияние многие языки, при разработке была цель сделать язык похожим на Java, но при этом лёгким для использования [непрограммистами](http://ru.wiktionary.org/wiki/%D0%BF%D0%BE%D0%BB%D1%8C%D0%B7%D0%BE%D0%B2%D0%B0%D1%82%D0%B5%D0%BB%D1%8C-%D0%BD%D0%B5%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%81%D1%82). Языком JavaScript не владеет какая-либо компания или организация, что отличает его от ряда языков программирования, используемых в веб-разработке.

Название «JavaScript» является зарегистрированным [товарным знаком](http://ru.wikipedia.org/wiki/%D0%A2%D0%BE%D0%B2%D0%B0%D1%80%D0%BD%D1%8B%D0%B9_%D0%B7%D0%BD%D0%B0%D0%BA) компании [Oracle Corporation](http://ru.wikipedia.org/wiki/Oracle_Corporation).

HTML– стандартный язык разметки документов во Всемирной паутине. Большинство веб-страниц содержат описание разметки на языке HTML (или XHTML). Язык HTML интерпретируется браузерами и отображается в виде документа в удобной для человека форме.

Язык HTML является приложением («частным случаем») SGML (стандартного обобщённого языка разметки) и соответствует международному стандарту ISO 8879.

Язык XHTML является более строгим вариантом HTML, он следует всем ограничениям XML и, фактически, XHTML можно воспринимать как приложение языка XML к области разметки гипертекста.

Во всемирной паутине HTML-страницы, как правило, передаются браузерам от сервера по протоколам HTTP или HTTPS, в виде простого текста или с использованием сжатия.

CSS – формальный язык описания внешнего вида документа, написанного с использованием языка разметки.

Преимущественно используется как средство описания, оформления внешнего вида веб-страниц, написанных с помощью языков разметки HTML и XHTML, но может также применяться к любым XML-документам, например, к SVG или XUL.

CSS используется создателями веб-страниц для задания цветов, шрифтов, расположения отдельных блоков и других аспектов представления внешнего вида этих веб-страниц. Основной целью разработки CSS являлось разделение описания логической структуры веб-страницы (которое производится с помощью HTML или других языков разметки) от описания внешнего вида этой веб-страницы (которое теперь производится с помощью формального языка CSS). Такое разделение может увеличить доступность документа, предоставить большую гибкость и возможность управления его представлением, а также уменьшить сложность и повторяемость в структурном содержимом. Кроме того, CSS позволяет представлять один и тот же документ в различных стилях или методах вывода, таких как экранное представление.

Visual Basic .NET (VB.NET) — это [объектно-ориентированный](http://ru.wikipedia.org/wiki/%D0%9E%D0%B1%D1%8A%D0%B5%D0%BA%D1%82%D0%BD%D0%BE-%D0%BE%D1%80%D0%B8%D0%B5%D0%BD%D1%82%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%BD%D0%BE%D0%B5_%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5) [язык программирования](http://ru.wikipedia.org/wiki/%D0%AF%D0%B7%D1%8B%D0%BA_%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D1%8F), который можно рассматривать как очередной виток эволюции [Visual Basic](http://ru.wikipedia.org/wiki/Visual_Basic) (VB), реализованный на платформе [Microsoft .NET](http://ru.wikipedia.org/wiki/Microsoft_.NET). VB.NET не имеет обратной совместимости с более ранней версией (Visual Basic 6.0).

Важнейшие изменения в VB.NET следующие.

Поддержка концепций [объектно-ориентированного программирования](http://ru.wikipedia.org/wiki/%D0%9E%D0%B1%D1%8A%D0%B5%D0%BA%D1%82%D0%BD%D0%BE-%D0%BE%D1%80%D0%B8%D0%B5%D0%BD%D1%82%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%BD%D0%BE%D0%B5_%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5)

с [конструкторами](http://ru.wikipedia.org/wiki/%D0%9A%D0%BE%D0%BD%D1%81%D1%82%D1%80%D1%83%D0%BA%D1%82%D0%BE%D1%80_(%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5)) и [деструкторами](http://ru.wikipedia.org/wiki/%D0%94%D0%B5%D1%81%D1%82%D1%80%D1%83%D0%BA%D1%82%D0%BE%D1%80_(%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5)), [наследованием](http://ru.wikipedia.org/wiki/%D0%9D%D0%B0%D1%81%D0%BB%D0%B5%D0%B4%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5_(%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5)), перекрытием методов (Overrides) и др. Есть даже переопределение знаков операций.

Компиляция в [байт-код](http://ru.wikipedia.org/wiki/%D0%91%D0%B0%D0%B9%D1%82-%D0%BA%D0%BE%D0%B4) (intermediate language, IL), исполняемый с помощью виртуальной машины common language runtime (CLR).

Использование всего набора объектных библиотек, входящих в .NET Framework, включающих мощные средства по работе с формами ([Windows Forms](http://ru.wikipedia.org/wiki/Windows_Forms)), базами данных ([ADO.NET](http://ru.wikipedia.org/wiki/ADO.NET)), графикой (GDI+), средствами обеспечения безопасности, веб-страницами ([ASP.NET](http://ru.wikipedia.org/wiki/ASP.NET)) и т. п.

Поддержка свободной многопоточности.

Поддержка [структурной обработки исключений](http://ru.wikipedia.org/wiki/%D0%A1%D1%82%D1%80%D1%83%D0%BA%D1%82%D1%83%D1%80%D0%BD%D0%B0%D1%8F_%D0%BE%D0%B1%D1%80%D0%B0%D0%B1%D0%BE%D1%82%D0%BA%D0%B0_%D0%B8%D1%81%D0%BA%D0%BB%D1%8E%D1%87%D0%B5%D0%BD%D0%B8%D0%B9) (structured exception handling, SEH).

С переходом на платформу .NET, изменилась сама [парадигма программирования](http://ru.wikipedia.org/wiki/%D0%9F%D0%B0%D1%80%D0%B0%D0%B4%D0%B8%D0%B3%D0%BC%D0%B0_%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D1%8F) (и это наиболее важное изменение).

Настройка сервера

Для настройки сервера использовалась программа PuTTY

PuTTY — свободно распространяемый клиент для различных протоколов удалённого доступа, включая SSH, Telnet, rlogin. Также имеется возможность работы через последовательный порт. Позволяет подключиться и управлять сервером. В PuTTY реализована только клиентская сторона соединения — сторона отображения, в то время как сама работа выполняется на другой стороне.

Программное обеспечение клиента

В качестве клиентской программы на компьютере пользователя выступает интернет браузер.

Список браузеров, поддержка которых гарантирована:

* Opera 12 и выше
* Mozilla Firefox 20.0 и выше
* Google Chrome 31 и выше
* Internet Explorer 9 и выше
* Safari 5 и выше

# 2 Специальная часть.

## 2.1 Проектирование базы данных.

### 2.1.1 Инфологическая модель предметной области.

Цель инфологического моделирования – обеспечение наиболее естественных для человека способов сбора и представления той информации, которую предполагается хранить в создаваемой базе данных. Поэтому инфологическую модель данных пытаются строить по аналогии с естественным языком, но последний не может быть использован в чистом виде из-за сложности компьютерной обработки текстов и неоднозначности любого естественного языка. Основными конструктивными элементами инфологических моделей являются сущности, связи между ними и их свойства (атрибуты).

*Сущность* – любой различимый объект (объект, который мы можем отличить от другого), информацию о котором необходимо хранить в базе данных. Сущностями могут быть люди, места, самолеты, рейсы, вкус, цвет и т.д. Необходимо различать такие понятия, как *тип сущности* и *экземпляр сущности*. Понятие тип сущности относится к набору однородных личностей, предметов, событий или идей, выступающих как целое. Экземпляр сущности относится к конкретной вещи в наборе. Например, типом сущности может быть город, а экземпляром – Москва, Киев и т.д.

*Атрибут* – поименованная характеристика сущности. Его наименование должно быть уникальным для конкретного типа сущности.

Для построения инфологической модели предметной области были выделены сущности предметной области, описание которых представлено в таблице 2.1, а также схема существующих потоков информации (Рисунок 2.1)

Таблица 2.1 – Описание сущностей предметной области.

|  |  |
| --- | --- |
| **Сущность** | **Определение** |
| Специалист | Сотрудники отдела информационных технологий и связи ОАО «СНХРС» |
| Задача | Заявка от пользователя или по служебной документации зарегистрированная для исполнения специалисту |
| Комментарий | Текстовое сообщение, оставленное специалистом для уточнения хода выполнения заявки |
| Подразделение | Наименование структурного подразделения предприятия |
| Тип проблемы | Утвержденная категория возможных проблем |
| Приоритет | Уровни важности решения задачи |

После определения атрибутов сущностей, определим основные связи между ними.

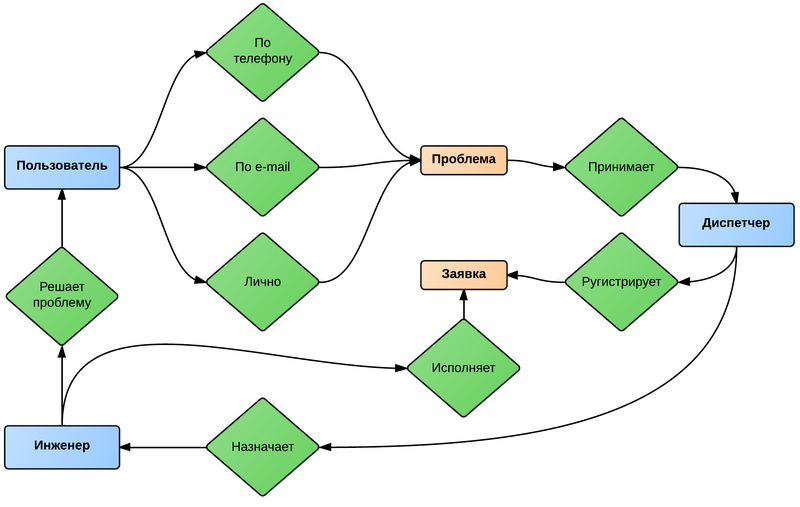


Рисунок 2.1 – Схема предметной области до внедрения АИС

Проанализировав существующую модель и определив возможные точки автоматизации была составлена модель информационных поток при использовании АИС (Рисунок 2.2)

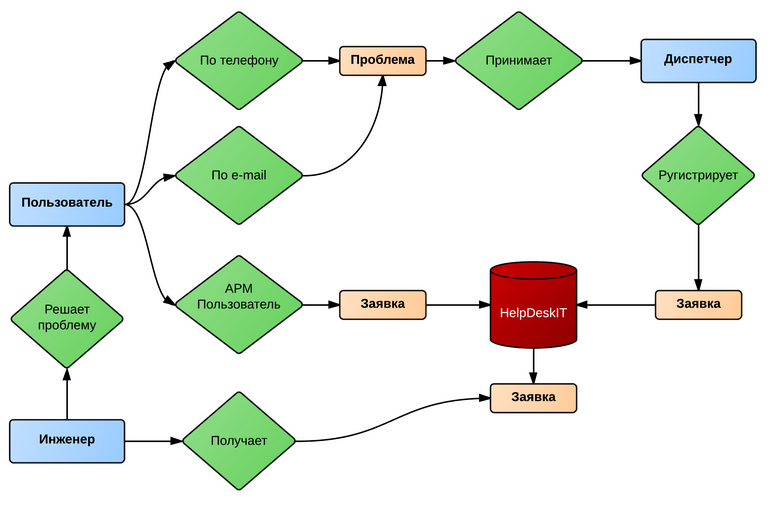


Рисунок 2.2 – Схема предметной области после внедрения АИС

Как видно из схему центром обработки информации является АИС аккумулируя в себе все данные, тем самым предоставляя большие возможности по обработке информации.

### 2.1.2 Даталогическая модель предметной области.

Даталогическая модель является моделью логического уровня и представляет собой отображение логических связей между элементами данных безотносительно к их содержанию и среде хранения. Эта модель строится в терминах информационных единиц, допустимых в той конкретной СУБД, в среде которой мы проектируем базу данных. Этап создания ДЛМ называется даталогическнм проектированием. Описание логической структуры базы данных на языке СУБД называется схемой.

На основе изучений предметной области и инфологической модели спроектируем таблицы будущей базы данных, атрибуты, типы данных.

Таблица hd\_user хранит информацию о сотрудниках, работающих с АИС

Таблица 2.2 – Таблица hd\_user

|  |  |  |
| --- | --- | --- |
| **Атрибут** | **Тип** | **Описание** |
| id | Счетчик | Уникальный код сотрудника |
| login | VARCHAR(255) | Логин сотрудника |
| password | VARCHAR(255) | Пароль сотрудника |
| fullname | VARCHAR(255) | Полное ФИО сотрудника |
| position | VARCHAR(255) | Должность |
| local | INT | Индикатор - локальный или сетевой пользователь |
| list | INT | Индикатор - будет ли фигурировать сотрудник в списках |
| active | INT | Индикатор – активен или не активен сотрудник |
| access\_user | INT | Параметры доступа к АРМ «Пользователь» |
| access\_disp | INT | Параметры доступа к АРМ «Диспетчер» |
| access\_ingener | INT | Параметры доступа к АРМ «Инженер» |
| access\_admin | INT | Параметры доступа к АРМ «Администратор» |

Таблица hd\_priority хранит информацию о приоритетах задачи

Таблица 2.3 – Таблица hd\_ priority

|  |  |  |
| --- | --- | --- |
| **Атрибут** | **Тип** | **Описание** |
| id | Счетчик | Уникальный код приоритета |
| priority | VARCHAR(255) | Наименование приоритета |

Таблица hd\_department хранит информацию о подразделениях

Таблица 2.4 – Таблица hd\_ department

|  |  |  |
| --- | --- | --- |
| **Атрибут** | **Тип** | **Описание** |
| id | Счетчик | Уникальный код подразделения |
| department | VARCHAR(255) | Наименование подразделения |

Таблица hd\_tip\_problem хранит информацию о типах проблем

Таблица 2.5 – Таблица hd\_tip\_problem

|  |  |  |
| --- | --- | --- |
| **Атрибут** | **Тип** | **Описание** |
| id | Счетчик | Уникальный код типа проблем |
| tip\_problem | VARCHAR(255) | Наименование типа проблем |

Таблица hd\_comment\_ticket хранит информацию о комментариях

Таблица 2.6 – Таблица hd\_comment\_ticket

|  |  |  |
| --- | --- | --- |
| **Атрибут** | **Тип** | **Описание** |
| id | Счетчик | Уникальный код комментария |
| id\_ticket | INT | Код задачи для которой оставлен комментарий |
| comment | VARCHAR(255) | Текст комментария |
| com\_date | DATATIME | Дата и время комментария |
| comment\_ing | VARCHAR(255) | Сотрудник оставивший комментарий |

Таблица hd\_ticket хранит информацию о задачах зарегистрированных в АИС

Таблица 2.7 – Таблица hd\_ticket

|  |  |  |
| --- | --- | --- |
| **Атрибут** | **Тип** | **Описание** |
| id | Счетчик | Уникальный код задачи |
| tip\_ticket | INT | Тип задачи |
| numsz | VARCHAR(255) | Номер служебной записки |
| problem | INT | Тип проблемы |
| depart | INT | Подразделение |
| fio | VARCHAR(255) | ФИО пользователя подавшего заявку |
| tel | VARCHAR(255) | Телефон пользователя подавшего заявку |
| ip | VARCHAR(255) | IP адрес пользователя подавшего заявку |
| status | INT | Статус задачи |
| deleted | INT | Индикатор - актуальная или удаленная задача |
| priority | INT | Приоритет задачи |
| date\_open | DATATIME | Дата открытия задачи |
| date\_close | DATATIME | Дата закрытия задачи |
| date\_mod | DATATIME | Дата изменения задачи |
| user\_open | INT | Сотрудник открывший задачу |
| user\_recipient | INT | Сотрудник получивший задачу |
| user\_mod | INT | Сотрудник изменивший заявку |
| content | TEXT | Текст задачи |

### 2.1.3 Физическая модель базы данных.

Физическая модель подразумевает построение базы данных с помощью выбранного СУБД.

Основными критериями выбора MySQL в качестве СУБД явились:

* гибкая система привилегий и паролей;
* гибкая поддержка форматов чисел, строк переменной длины и меток времени;
* интерфейс с языками C и Perl, PHP;
* быстрая работа, масштабируемость;
* бесплатна;
* хорошая поддержка со стороны провайдеров услуг хостинга;

Самым распространенным средством управления MySQL является бесплатная система phpMyAdmin

Для создания базы данных имя базы и выбрать кодировку в соответствующих полях, указанных красным цветом на рисунке 5.

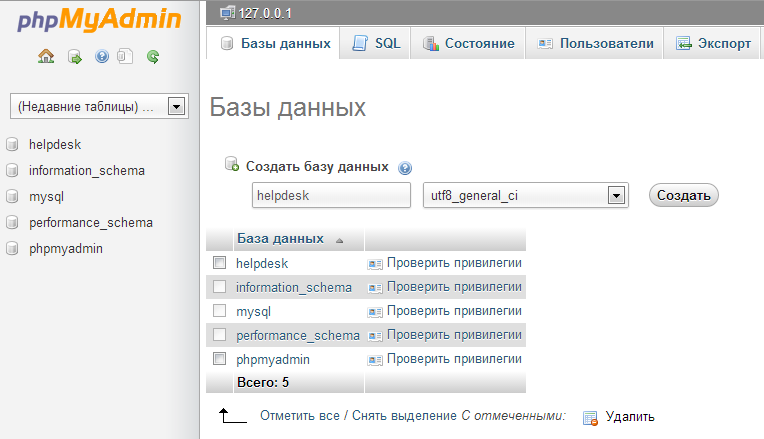


Рисунок 2.3 – Создание базы данных

После создания базы данных необходимо создать нужное количество таблиц, ввести название и количество столбцов, как показано на рисунке 6.

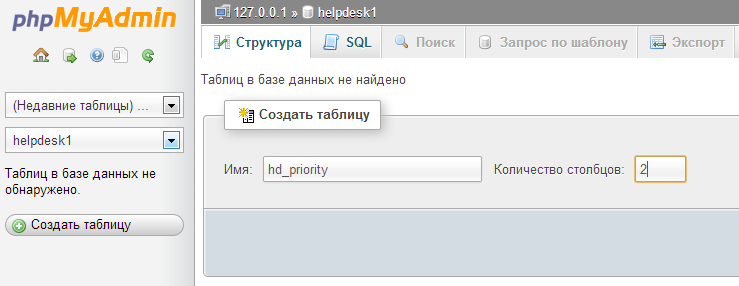


Рисунок 2.4 – Создание таблицы базы данных

После создания таблицы добавляем названия столбцов и тип хранимых данных.

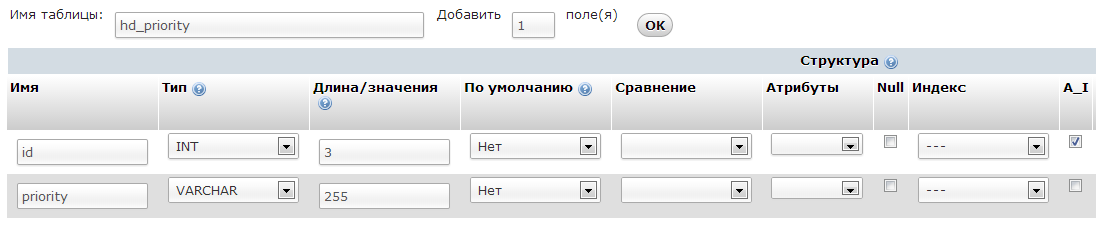


Рисунок 2.4 – Определение название столбцов и типов данных.

Аналогичным образом создаем все таблицы спроектированные в даталогической модели

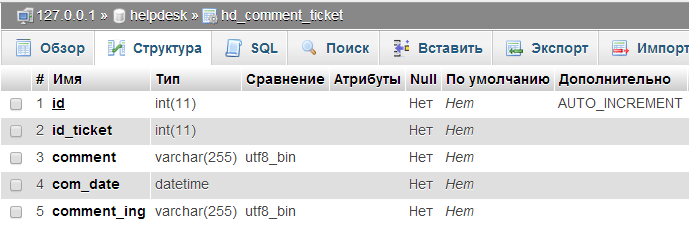


Рисунок 2.5 – Структура таблицы hd\_comment\_ticket в PhpMyAdmin

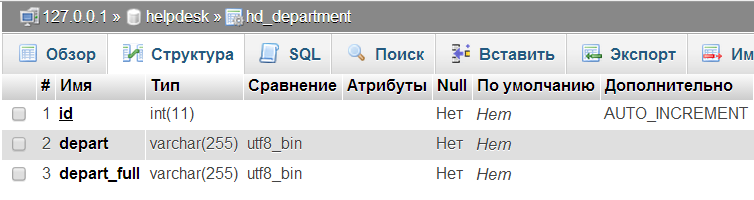


Рисунок 2.6 – Структура таблицы hd\_department в PhpMyAdmin

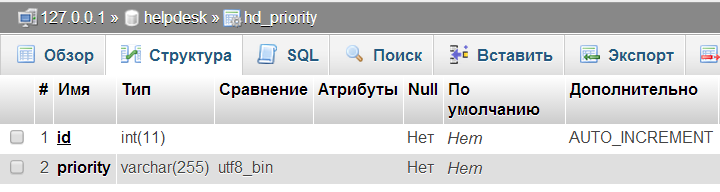


Рисунок 2.7 – Структура таблицы hd\_priority в PhpMyAdmin

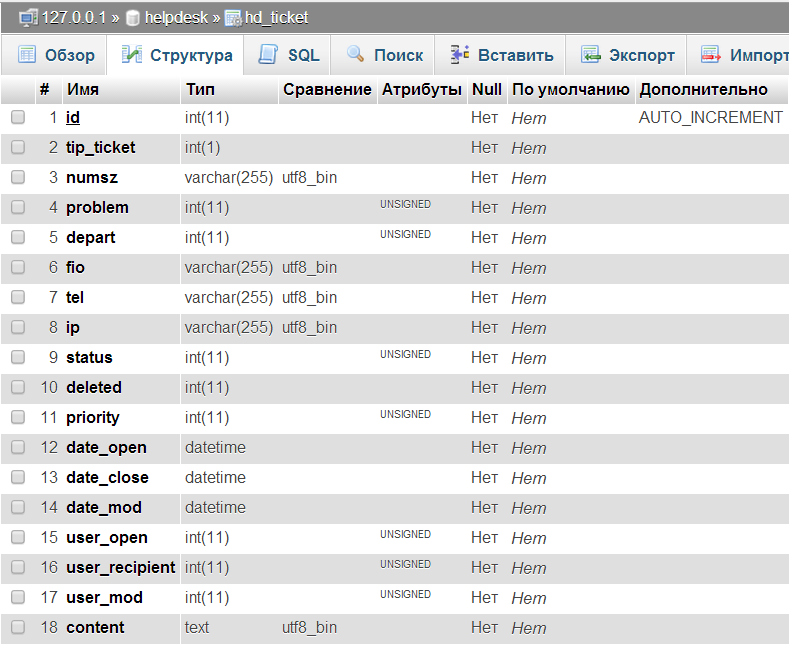


Рисунок 2.8 – Структура таблицы hd\_ticket в PhpMyAdmin

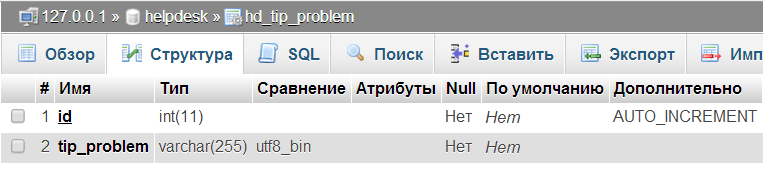


Рисунок 2.9 – Структура таблицы hd\_tip\_problem в PhpMyAdmin

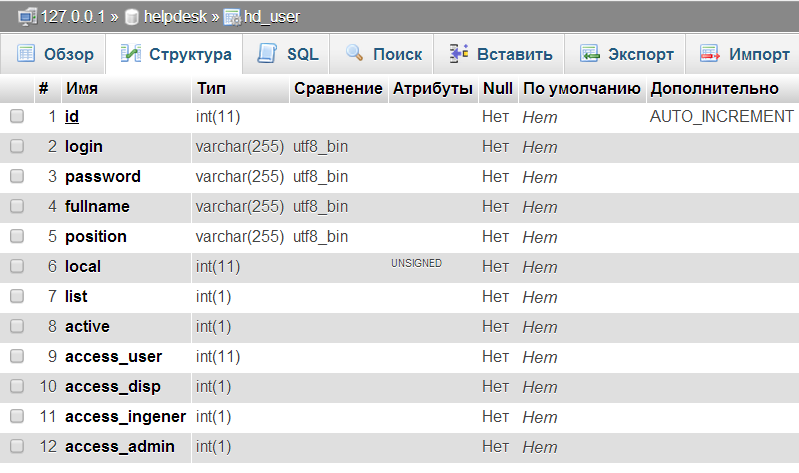


Рисунок 2.10 – Структура таблицы hd\_user в PhpMyAdmin

## 2.2 Реализация программной оболочки для базы данных.

### 2.2.1 Описание процесса и инструментов разработки.

Основным направление разработки стало создание Web-приложения, так как была необходимость мобильность, работа пользователей без установки дополнительного программного обеспечения и возможность работы с Windows и Linux среде.

Языком программирования обрабатывающий сценарии на сервере был выбран PHP, на клиентском компьютере JavaScript.

Для формирования внешнего вида приложения использовались HTML и CSS

В качестве основным инструмента написания программного кода был использован Notepad++.

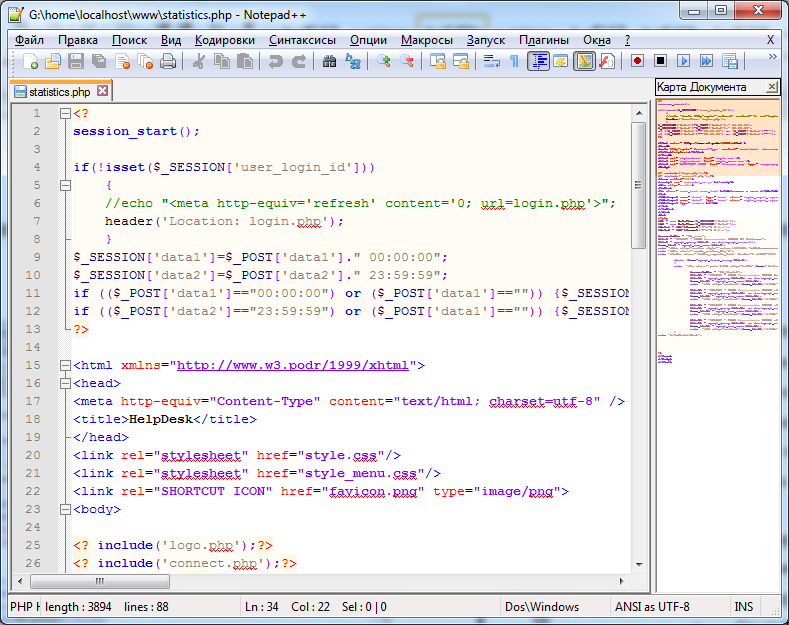


Рисунок 2.11 – Редактор программного кода Notepad++

Редактор специально создан для набора текстов с помощью языков программирования, имеет наипростейший интерфейс, огромный функционал. Основан на компоненте Scintilla и имеет лицензию GPL

Текстовый редактор состоит практически только из одного окна, в панель инструментов выложены все самые необходимые функции по работе с текстом, а ниже расположена панель вкладок, в каждой из которой можно расположить части программных кодов или готовый шаблон.

Основные возможности:

* подсветка текста и сворачивание блоков, следуя правилам синтаксиса языка программирования;
* WYSIWYG - своеобразный графический редактор текста, знаком создателям сайтов;
* настройка режима подсветки текста;
* автоматическое завершение слова позволяет прописывать только часть кода, более подробно можно прочитать на сайте разработчика;
* работа сразу с несколькими документами, благодаря системе вкладок;
* одновременный просмотр двух документов, разрезав рабочее окно на две половины;
* поддержка постоянной замены выражений с помощью поиска;
* Drag'n'Drop поможет открыть документ, передвинув иконку документа в редактор и также легко перебрасывать текст внутри программы;
* динамическое изменение окон просмотра;
* при удалении или изменении когда-либо используемого файла в программе появляется уведомление при его новом открытии;
* увеличение или уменьшение (функция Scintill);
* поддержка мультиязычности, база языков пополняется любителями, которые переводят программу на свой родной язык;
* добавление быстрых заметок (Ctrl+F2);
* выделение скобок для определения начала и конца блока;
* запись макроса и его выполнение

Огромным достоинством является расширенная поддержка большого числа синтаксисов: C, C++, Java, C#, XML, HTML, PHP, CSS, makefile, ASCII art (.nfo), doxygen, ini file, batch file, Javascript, ASP, VB/VBS, SQL, Objective-C, RC resource file, Pascal, Perl, Python, Lua, TeX, TCL, Assembler, Ruby, Lisp, Scheme, Properties, Diff, Smalltalk, Postscript, VHDL, Ada, Caml, AutoIt, KiXtart, Matlab, Verilog, Haskell, InnoSetup, CMake, YAML

Структура приложения имеет модульное исполнение. Каждая АРМ и основные логические части программы выполнены в виде отдельного PHP файла. Таким образом, каждая часть автономна и при необходимости ее можно дорабатывать, не вмешиваясь в работу остальным частей.

Данный подход позволяет постепенно модернизировать каждую часть АИС, не приводя к долговременному простою при обнаружении ошибок в работе, так как проблема будет локально влиять только на свою часть.

Для создания приложения для Windows был использован язык программирования VB.NET. Среда программирования – Microsoft Visual Studio 2013.

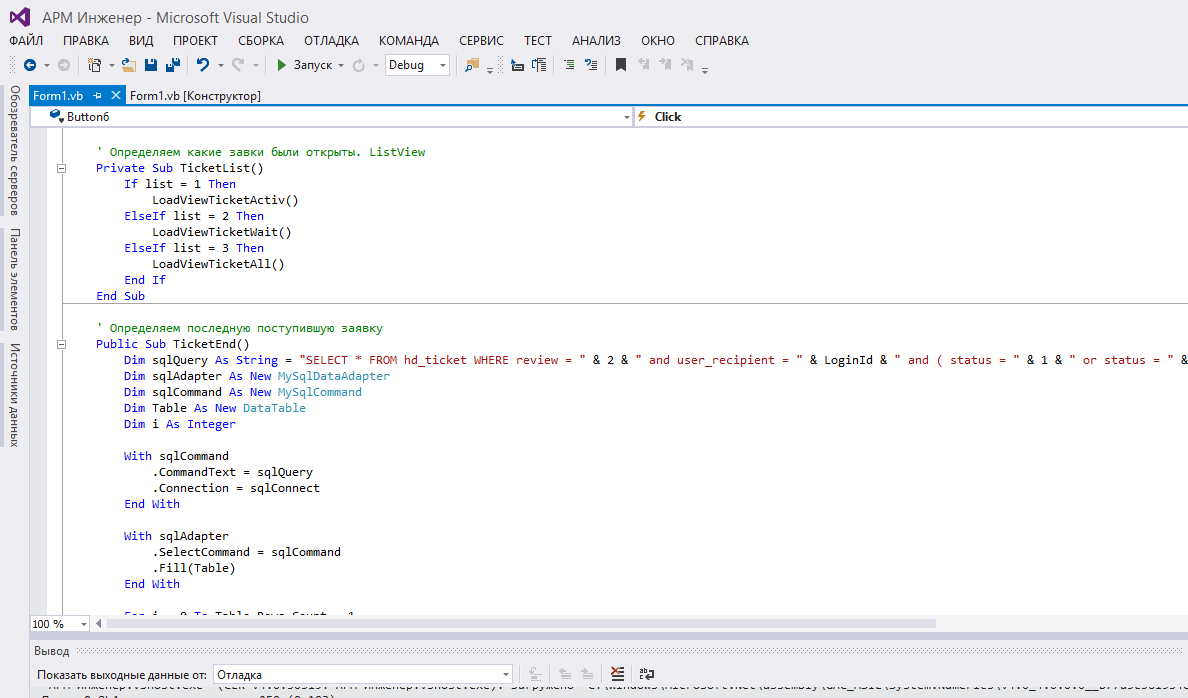


Рисунок 2.12 –Часть кода на Visual Studio 2013

Visual Basic .NET — это [объектно-ориентированный](http://ru.wikipedia.org/wiki/%D0%9E%D0%B1%D1%8A%D0%B5%D0%BA%D1%82%D0%BD%D0%BE-%D0%BE%D1%80%D0%B8%D0%B5%D0%BD%D1%82%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%BD%D0%BE%D0%B5_%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5) [язык программирования](http://ru.wikipedia.org/wiki/%D0%AF%D0%B7%D1%8B%D0%BA_%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D1%8F), который можно рассматривать как очередной виток эволюции  [Visual Basic](http://ru.wikipedia.org/wiki/Visual_Basic) (VB), реализованный на платформе [Microsoft .NET](http://ru.wikipedia.org/wiki/Microsoft_.NET). VB.NET не имеет обратной совместимости с более ранней версией (Visual Basic 6.0). Развитие проектов старых версий (\*.vbp) возможно только после предварительной конвертации их в формат VB.NET специальным мастером (Migration Wizard); однако, как правило, после конвертации требуется существенная ручная доработка текстов.

Microsoft Visual Studio — линейка продуктов компании [Майкрософт](http://ru.wikipedia.org/wiki/%D0%9C%D0%B0%D0%B9%D0%BA%D1%80%D0%BE%D1%81%D0%BE%D1%84%D1%82), включающих [интегрированную среду разработки](http://ru.wikipedia.org/wiki/%D0%98%D0%BD%D1%82%D0%B5%D0%B3%D1%80%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%BD%D0%B0%D1%8F_%D1%81%D1%80%D0%B5%D0%B4%D0%B0_%D1%80%D0%B0%D0%B7%D1%80%D0%B0%D0%B1%D0%BE%D1%82%D0%BA%D0%B8) программного обеспечения и ряд других инструментальных средств. Данные продукты позволяют разрабатывать как [консольные](http://ru.wikipedia.org/wiki/%D0%9A%D0%BE%D0%BD%D1%81%D0%BE%D0%BB%D1%8C%D0%BD%D0%BE%D0%B5_%D0%BF%D1%80%D0%B8%D0%BB%D0%BE%D0%B6%D0%B5%D0%BD%D0%B8%D0%B5) [приложения](http://ru.wikipedia.org/wiki/%D0%9F%D1%80%D0%B8%D0%BA%D0%BB%D0%B0%D0%B4%D0%BD%D0%BE%D0%B5_%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%BD%D0%BE%D0%B5_%D0%BE%D0%B1%D0%B5%D1%81%D0%BF%D0%B5%D1%87%D0%B5%D0%BD%D0%B8%D0%B5), так и приложения с [графическим интерфейсом](http://ru.wikipedia.org/wiki/%D0%93%D1%80%D0%B0%D1%84%D0%B8%D1%87%D0%B5%D1%81%D0%BA%D0%B8%D0%B9_%D0%B8%D0%BD%D1%82%D0%B5%D1%80%D1%84%D0%B5%D0%B9%D1%81_%D0%BF%D0%BE%D0%BB%D1%8C%D0%B7%D0%BE%D0%B2%D0%B0%D1%82%D0%B5%D0%BB%D1%8F), в том числе с поддержкой технологии [Windows Forms](http://ru.wikipedia.org/wiki/Windows_Forms), а также [веб-сайты](http://ru.wikipedia.org/wiki/%D0%A1%D0%B0%D0%B9%D1%82), [веб-приложения](http://ru.wikipedia.org/wiki/%D0%92%D0%B5%D0%B1-%D0%BF%D1%80%D0%B8%D0%BB%D0%BE%D0%B6%D0%B5%D0%BD%D0%B8%D0%B5), [веб-службы](http://ru.wikipedia.org/wiki/%D0%92%D0%B5%D0%B1-%D1%81%D0%BB%D1%83%D0%B6%D0%B1%D0%B0) как в [родном](http://ru.wikipedia.org/wiki/%D0%9C%D0%B0%D1%88%D0%B8%D0%BD%D0%BD%D1%8B%D0%B9_%D0%BA%D0%BE%D0%B4), так и в [управляемом](http://ru.wikipedia.org/wiki/%D0%A3%D0%BF%D1%80%D0%B0%D0%B2%D0%BB%D1%8F%D0%B5%D0%BC%D1%8B%D0%B9_%D0%BA%D0%BE%D0%B4) кодах для всех платформ, поддерживаемых [Microsoft Windows](http://ru.wikipedia.org/wiki/Microsoft_Windows), [Windows Mobile](http://ru.wikipedia.org/wiki/Windows_Mobile), [Windows CE](http://ru.wikipedia.org/wiki/Windows_CE), [.NET Framework](http://ru.wikipedia.org/wiki/.NET_Framework), [Xbox](http://ru.wikipedia.org/wiki/Xbox), [Windows Phone](http://ru.wikipedia.org/wiki/Windows_Phone) [.NET Compact Framework](http://ru.wikipedia.org/wiki/.NET_Compact_Framework) и [Microsoft Silverlight](http://ru.wikipedia.org/wiki/Microsoft_Silverlight)

### 2.2.2 Описание структуры и программных модулей АИС.

Таблица 2.8 – Описание программных модулей.

| **Программный модуль** | **Функционал программного модуля** |
| --- | --- |
| add\_user.php | Добавление/редактирование пользователя |
| admin\_department.php | Изменение списка подразделений |
| admin\_settings.php | Системные настройки АИС для подключения к серверу БД |
| admin\_tip\_problem.php | Изменение списка типов проблем |
| arm\_admin.php | Основное окно АРМ «Администратор» |
| arm\_disp.php | Основное окно АРМ «Диспетчер» |
| arm\_ingener.php | Основное окно АРМ «Инженер» |
| arm\_user.php | Основное окно АРМ «Пользователь» |
| auth.php | Программный код для авторизации пользователей |
| clear.php | Очистка переменных форм |
| config.ini | Файл настроек подключения к серверу БД |
| connect.php | Подключение к базу данных |
| index.php | Окно выбора АРМ |
| ldap.php | Авторизация пользователя через протокол LDAP |
| list\_depart.php | Вывод списка подразделений |
| list\_priority.php | Вывод списка приоритета |
| list\_problem.php | Вывод списка типов проблем |
| list\_status.php | Вывод списка статусов |
| list\_user.php | Вывод списка сотрудников |
| list\_user\_noselect.php | Вывод списка сотрудников без выбора |
| login.php | Окно ввода логина и пароля |

Продолжение таблицы 2.8

| **Программный модуль** | **Функционал программного модуля** |
| --- | --- |
| logo.php | Логотип и меню |
| logoff.php | Выход из приложения |
| mod\_poisk.php | Модуль быстрого поиска |
| search.php | Окно расширенного поиска |
| settings\_user.php | Настройка и список пользователей |
| stat\_disp.php | Вывод статистики для диспетчера |
| stat\_ingener.php | Вывод статистики для инженера |
| stat\_oper.php | Вывод статистики для оперативки |
| z.php | Окно подробной информации по заявке |
| statistics.php | Статистика заявок по всем сотрудникам |

## 2.3 Разработка руководства пользователя.

### 2.3.1 Область применения.

АИС предназначена для применения в ОАО «Салаватнефтехимремстрой» и выполняет функции по автоматизации приема, учета и обработки заявок в IT службу предприятия

### 2.3.2 Уровень подготовки пользователя.

Пользователи АРМ ОТ должны обладать квалификацией, обеспечивающей, как минимум:

* базовые навыки работы на персональном компьютере с графическим пользовательским интерфейсом (клавиатура, мышь, управление окнами и приложениями, файловая система);
* базовые навыки использования стандартной клиентской программы (Microsoft Internet Explorer 6.0 и выше) в среде Интернета (настройка подключений, доступ к веб-сайтам, навигация, формы и другие типовые интерактивные элементы);
* базовые навыки работы с офисной оргтехникой (принтер, сканер, копировальный аппарат, факс)
* базовые навыками использования стандартной почтовой программы (создание, отправка и получение e-mail);
* знание основ информационной безопасности;
* знание принципов организации процесса подготовки и публикации материалов

### 2.3.3 Назначение системы

Разработанная автоматизированная информационная система необходима для учета и обработки заявок в отдел информационных технологий и связи ОАО «Салаватнефтехимремстрой».

### 2.3.4 Требования к программному обеспечению

Для работы с АИС пользователь должен использовать интернет браузер Internet Explorer 8.0 и выше, Mozila FireFox 10 и выше, Google Chrome 15 и выше.

### 2.3.5 Требования к аппаратному обеспечения

Для корректной работыАИС требуется следующее оборудование:

* Intel Core i3 2.2 Ghz или производительнее
* 2 Gb оперативной памяти.
* 15 Gb на жестком диске.
* SVGA видео карта
* сетевая карта не менее 100 Мбит/с

### 2.3.6 Установка и запуск приложения

Работа пользователя ведется через web-приложение, которое не требует предварительной установки каких-либо компонентов АИС на персональный компьютер.

Для того чтобы запустить web-приложение АИС необходимо в адресной строке браузера ввести URL - адрес и нажать «Enter».

### 2.3.7 Руководство АРМ «Пользователь».

Запуск приложения.

Для запуска приложения необходимо открыть браузер и ввести адрес страницы. Данный адрес устанавливается администратором системы. Пример адреса – <https://helpdesk.snhrs.ru/user>

Откроется главное окно АРМ «Пользователь»

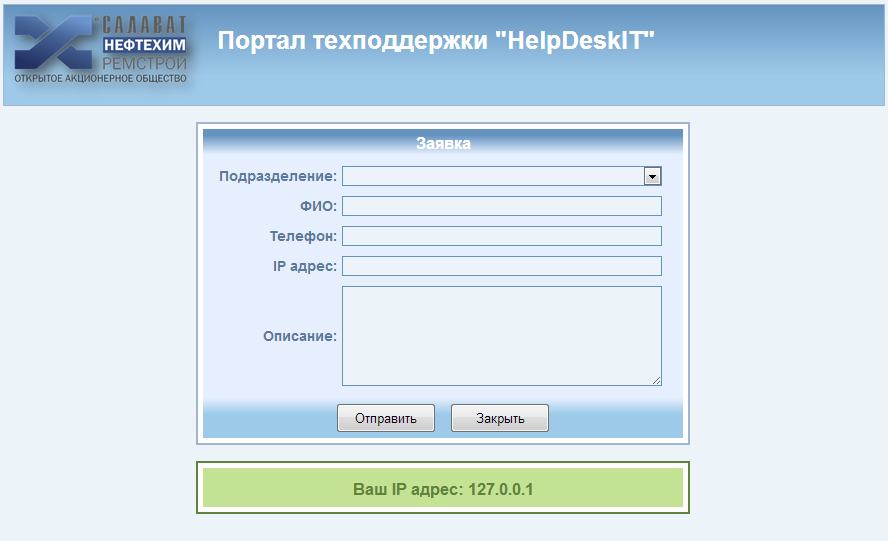


Рисунок 2.13 Окно АРМ «Пользователь»

В данном окне необходимо заполнить соответствующие поля и нажать кнопку «Отправить». После чего получаем подтверждение регистрации заявки окном оповещения

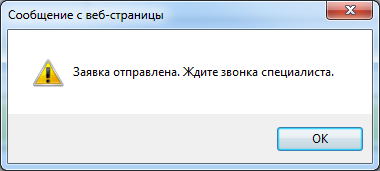


Рисунок 2.14 Окно оповещения о регистрации заявки.

В случае не заполнения одного из обязательных полей получаем сообщение об этом с название пустого поля.

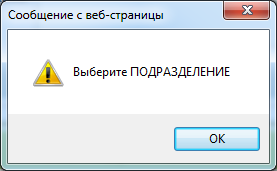


Рисунок 2.15 Сообщение о пустом поле.

### 2.3.8 Руководство АРМ «Инженер», WEB-приложение

Запуск приложения.

Для запуска приложения необходимо открыть браузер и ввести адрес страницы авторизации. Данный адрес устанавливается администратором системы. Пример адреса – <https://helpdesk.snhrs.ru/>

Авторизация.

Открывается окно авторизации, в котором необходимо ввести полученные логин и пароль.

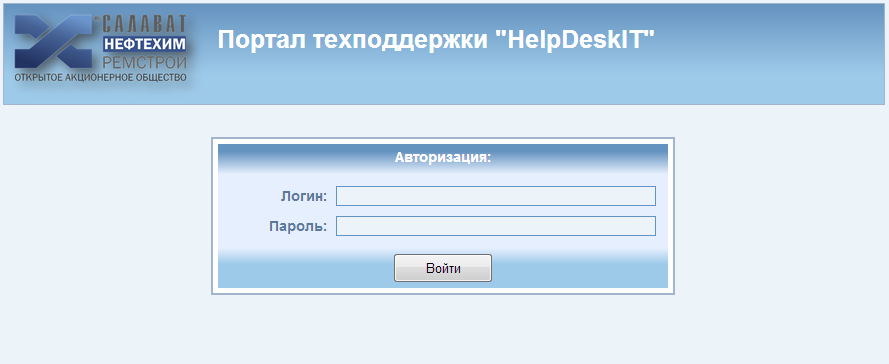


Рисунок 2.16 Окно авторизации

При вводе неверных данных авторизации система отображает сообщение.

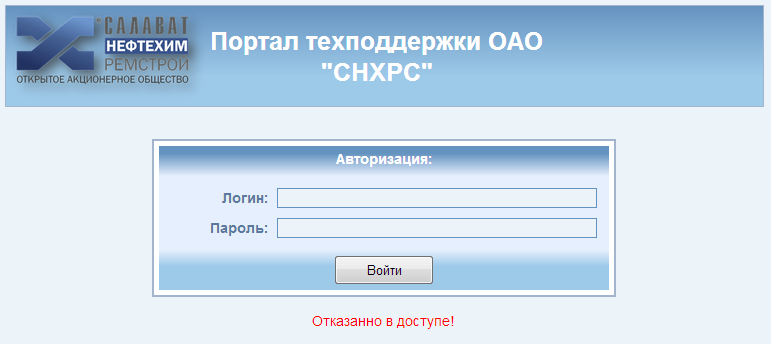


Рисунок 2.17 Ошибка авторизации

После успешной авторизации открывается главная форма выбора необходимой АРМ.

Над иконками каждой системы имеет индикатор доступа

* - доступ к АРМ предоставлен
* - доступ к АРМ отсутствует

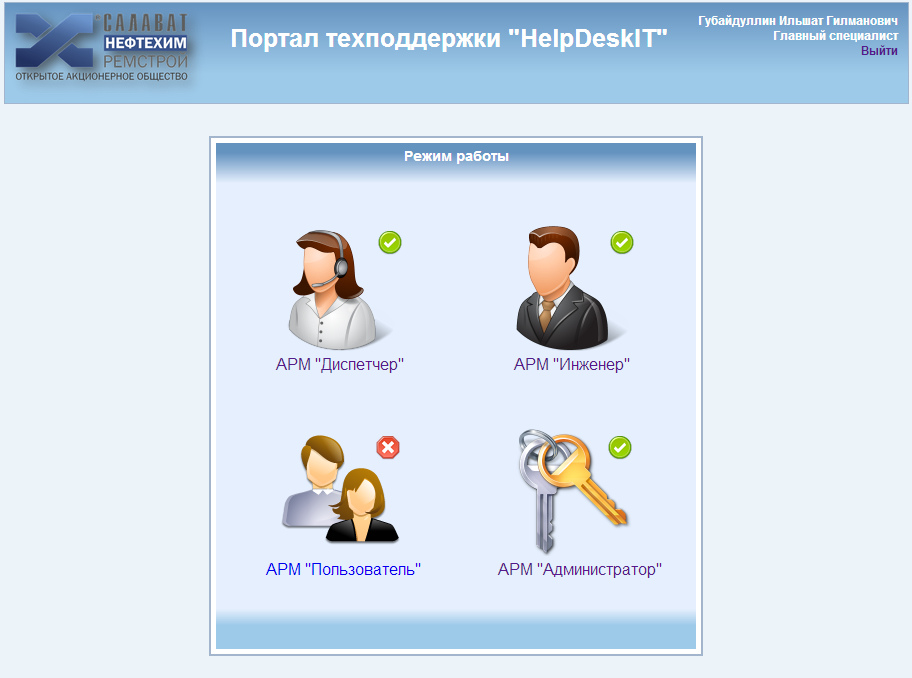


Рисунок 2.18 Окно выбора АРМ

Выбрав АРМ «Инженер» видим основное рабочее место инженера.

Оно состоит из нескольких функциональных блоков.

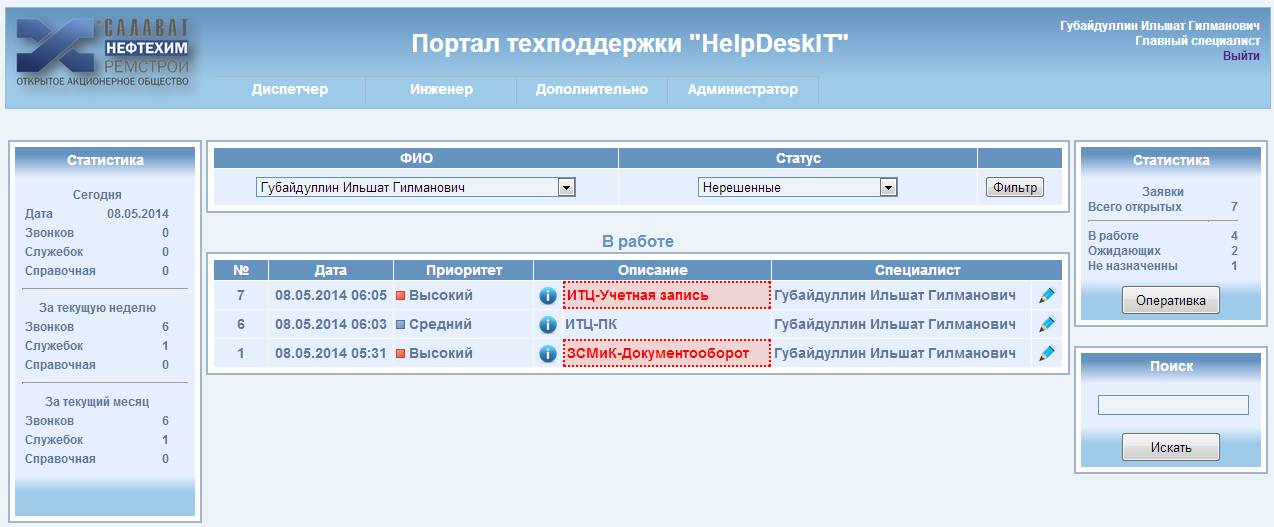


Рисунок 2.19 Общий вид АРМ «Инженер»

Сверху располагается логотип, название АИС, информация о пользователе и меню системы. Под информацией о пользователе располагается ссылка «Выйти» для выхода из режима текущего пользователя.

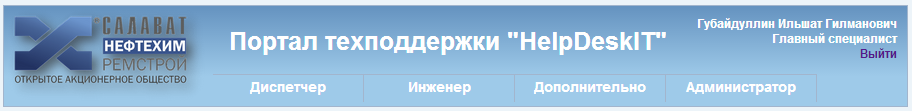


Рисунок 2.20 Логотип, название АИС, информация о пользователе, меню.

В центральной части располагается статистика по периодам текущий день, за текущую неделю и за текущий месяц.



Рисунок 2.21 Статистика по периодам.

Правая часть несет в себе информацию о текущем количестве заявок с разделением по статусам (Рисунок 36)

А так же модуль быстрого поиска по номеру заявки (Рисунок 37)

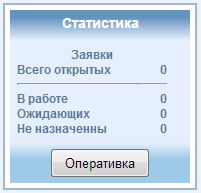


Рисунок 2.22 Информация о текущем количестве заявок.

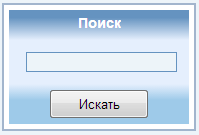


Рисунок 2.23 Модуль быстрого поиска по номеру заявки

Центральная часть отображается информация о текущих заявках специалиста. Они разделены на блоки по статусу:

* в работе
* ожидающие
* закрыто

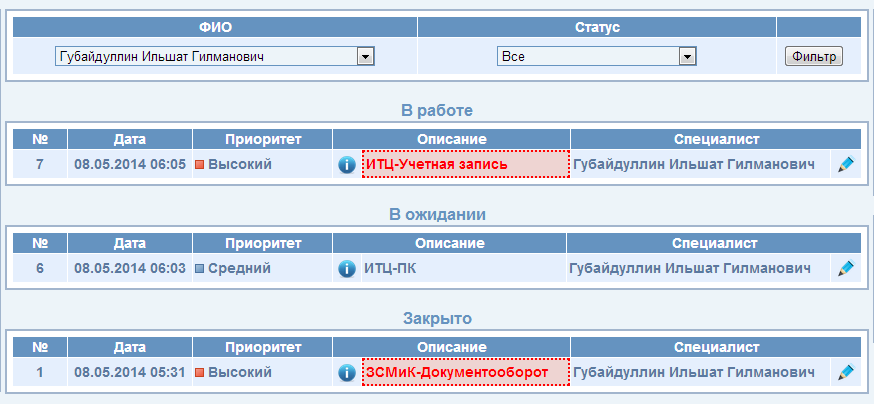


Рисунок 2.24 Информация о текущих заявках специалиста

Информация в каждом блоке:

* номер
* дата регистрации заявки
* приоритет, присвоенный задаче. Если приоритет высокий, то задача выделяется красным цветом
* краткое описание в формате подразделение – тип проблемы
* специалист, получивший задачу на исполнение

Специалист имеет возможность редактировать свои заявки. Для этого необходимо нажать кнопку в виде иконки карандаша G:\home\localhost\www\img\edit.png, после чего откроется окно редактирования заявки

Есть возможность изменить статус заявки соответствующими кнопками, изменить все данные самой заявки, а так же оставить комментарий.

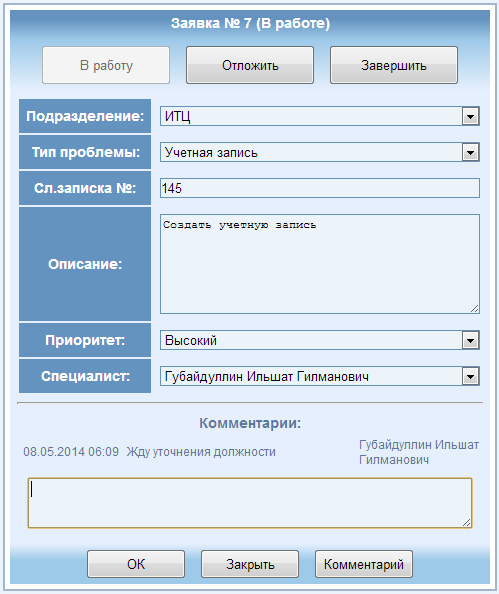


Рисунок 2.25 Окно редактирования заявки

### 2.3.8 Руководство АРМ «Инженер», Windows-приложение

Запуск приложения.

Для запуска приложения необходимо запустить АРМ Инженер.exe

Авторизация.

Открывается окно уже авторизованное окно. Программа запускается под тем пользователем, под которым вошли в систему. Рассматривать будем под пользователем 01gig – Губайдуллин Ильфат Гилманович.

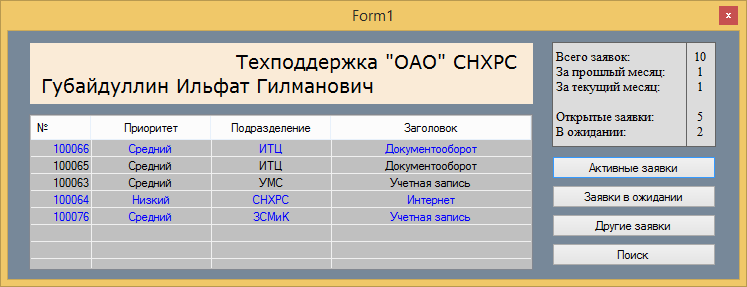


Рисунок 2.28 Основное окно.

Данное окно включает в себя 3 основных блока это:

* список открытых заявок
* статистика сотрудника
* кнопки для просмотра заявок с разными статусами: активная, в ожидании, все заявки



Рисунок 2.29 Список заявок

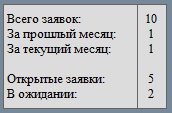


Рисунок 2.30 Статистика сотрудника

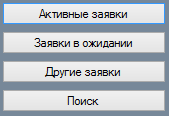


Рисунок 2.31 Кнопки для переключения

В списке заявок есть 2 типа цвета текста:

1. Черный – заявка, которая уже была просмотрена инженером.
2. Синий – вновь поступившая заявка, либо заявка еще не просмотренная инженером. Для того что бы заявку сделать просмотренным, необходимо два раза щелкнуть по нужной заявке.



Рисунок 2.32 Просмотренные заявки



Рисунок 2.33 Вновь поступившие заявки.

Приоритеты так же отличаются друг от друга в списке заявок. Чем выше приоритет, тем строка принимает более темно красный оттенок.

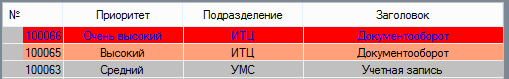


Рисунок 2.34 Отличия приоритетов.

При нажатии на заявку, основное окно расширяется и становиться более информативным.

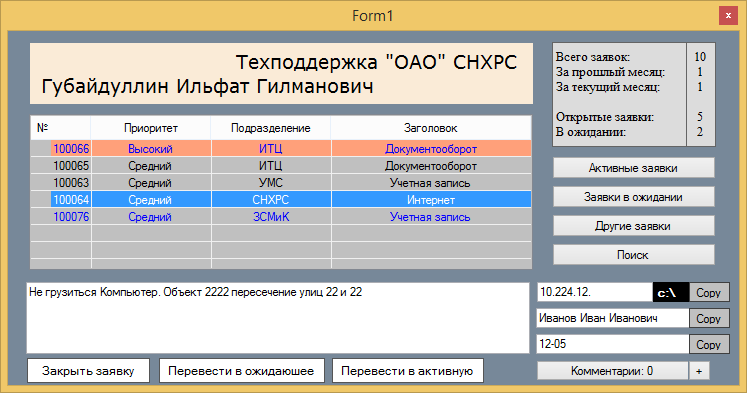


Рисунок 2.35 Расширенное окно заявок.

Составная часть, расширенной формы:

* описание заявки
* IP адрес пользователя
* ФИО пользователя
* контактный телефон пользователя
* кнопка для перевода заявки в ожидание, перевода заявки в активную и кнопка для того что бы закрыть заявку.
* кнопка для просмотра комментариев и кнопка для добавления комментария.

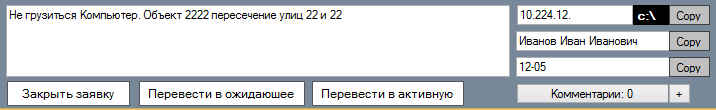


Рисунок 2.36 Составная часть расширенной формы

Приложение разработана таким образом, что при выполнении инженером заявки он выполнял минимум действий. Для облегчения роботы были введены такие кнопки как  и 

 - Кнопа PING. Предназначан для быстрого доступа к утилите ping. Для проверки соединения с узлом, в данном случае с компьютером по IP адресу.

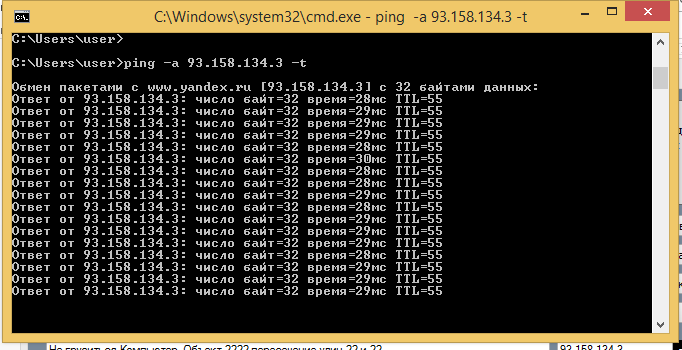


Рисунок 2.37 Утилита PING

# 3 Управление производственными и коммерческими операциями.

## 3.1 Исходные данные для расчетов.

Нормативные показатели экономической деятельности, необходимые для оценки эффективности внедрения, представлены в таблице 3.1.

Таблица 3.1 – Нормативные показатели

| Наименование показателя | Условное обозначение | Ед. изм. | Значение показателя |
| --- | --- | --- | --- |
| Коэффициент дополнительной заработной платы | kД | – | 0,25 |
| Плановый годовой фонд рабочего времени | Фр.в | час. | 1943 |
| Число рабочих дней (ср.значение) | ЧРД | дней | 20,33 |
| Количество рабочих смен | Ксм | – | 1 |
| Продолжительность рабочей смены | Тсм | час. | 8 |
| Уральский коэффициент | kУ | – | 0,15 |
| Отчисления во внебюджетные фонды | kОВФ | – | 30,2% |
| Пенсионный фонд | kпф | – | 22% |
| Фонд медицинского страхования | kфмс | – | 5,1% |
| Фонд социального страхования | kфсс | – | 2,9% |
| Страхование от несчастных случаев | kснс | – | 0,2% |
| Нормативный срок окупаемости капитальных вложений | Тн | лет | 3 |

Трудоемкость процесса разработки и внедрения ИС представлена в таблице 3.2. Данные получены путем хронометража.

Таблица 3.2 – Затраты времени на разработку и внедрение ИС

| Этап создания | Единица измерения | Значение всего | В т.ч. машинное время |
| --- | --- | --- | --- |
| Обследование объекта  автоматизации | час | 96 | 14 |
| Анализ и уточнение требований | час | 69 | 15 |
| Разработка технического задания | час | 70 | 70 |
| Проектирование | час | 126 | 126 |
| Программная реализация | час | 120 | 120 |
| Тестирование и отладка | час | 34 | 34 |
| ИТОГО на разработку | час | 515 | 379 |

Данные, влияющие на затраты при разработке и внедрении ИС, приведены в таблице 3.3.

Таблица 3.3 – Данные для расчета затрат на разработку и внедрение ИС

|  | Наименование показателя | Условное обозначение | Единица измерения | Значение показателя |
| --- | --- | --- | --- | --- |
| Общие  факторы | Оклад разработчика | Ор | руб. | 25000 |
| Количество разработчиков | Чр | чел. | 2 |
| Разработка | Количество ЭВМ | n | шт. | 2 |

Материалы, необходимые для разработки и внедрения ИС представлены в таблице 3.4.

Таблица 3.4 – Перечень материалов для разработки и внедрения ИС

|  |  |  |  |
| --- | --- | --- | --- |
| Наименование | Ед. измерения | Норма расхода, γi | Цена, руб./ед. |
| Разр. |
| Бумага | шт. | 1 | 150 |
| Ручка шариковая | шт. | 2 | 20 |

Трудоемкость работ в расчете на сотрудников, автоматизируемых разрабатываемой ИС

Таблица 3.5 – Расчет трудоемкости работ до внедрения системы. Диспетчер.

|  |  |
| --- | --- |
| Вид работы | Мин/мес |
| Прием звонка/опрос | 3150 |
| Оформление заявки | 1050 |
| Анализ проблемы, определение специалиста | 1050 |
| Звонок специалисту, описание проблемы | 3150 |
| Итого | 8400 |

Таблица 3.6 – Расчет трудоемкости работ после внедрения системы. Диспетчер.

|  |  |
| --- | --- |
| Вид работы | Мин/мес |
| Анализ проблемы | 525 |
| Назначение задачи специалисту | 525 |
| Итого | 1050 |

Таблица 3.7 – Расчет трудоемкости работ до внедрения системы. Инженер.

|  |  |
| --- | --- |
| Вид работы | Мин/мес |
| Прием звонка | 315 |
| Фиксирование задачи, анализ проблемы | 105 |
| Планирование процесса работы | 105 |
| Итого | 525 |

Таблица 3.8 – Расчет трудоемкости работ после внедрения системы. Инженер.

|  |  |
| --- | --- |
| Вид работы | Мин/мес |
| Планирование процесса работы | 105 |
| Итого | 105 |

Таблица 3.9 – Заработная плата сотрудников.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Наименование должности | Ед. изм. | Обозначение | Значение | |
| До внедрения АИС | После внедрения АИС |
| Диспетчера | руб. | Ом | 14500 | |
| Инженера | руб. | Ои | 20000 | |

## 3.2 Расчет капитальных затрат.

Капитальные затраты рассчитываются как сумма стоимости оборудования и затрат на разработку проектного решения:

К= Коб + Крпр (3.1)

где Коб - стоимость нового оборудования;

Крпр - затрат на разработку проектного решения;

Расчет стоимости оборудования:

Капитальные затраты на новое оборудование рассчитывается как сумма затрат на отдельные виды оборудования.

 (3.2)

где Цi- цена оборудования в руб.;

Сi- количество оборудования;

Для определения стоимости оборудования (Коб) необходимо составить спецификацию.

Таблица 3.10 – Спецификация на оборудование

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Наименование оборудования | ед. изм. | количество | Стоимость, руб. | |
| Сервер HP ProLiant DL360p Intel Xeon E5-2630v2 2.6GHz 15MB 16Gb Platunum 460W Gen8 Base EU Svr 1U | шт | 1 | 208680 | 208680 |
| 600VA Ippon Back Power Pro 600 New | шт | 1 | 2466 | 2466 |
| Итого стоимость оборудования | | | | 211146 |
| Капитальные затраты на оборудование | | | | 211146 |

Получаем, что Коб=211146 руб.

Затраты на разработку проектного решения рассчитываются по формуле:

КПРП = ЗФТОР + ЗОВФ + ЗЭВМ + ЗСПС + ЗК (3.3)

где

Крпр - затрат на разработку проектного решения;

*ЗФОТР* – общий фонд оплаты труда разработчиков, руб.;

*ЗОВФ* – отчисления во внебюджетные фонды с заработной платы разработчиков, руб.;

*ЗЭВМ* – затраты, связанные с эксплуатацией техники, руб.;

*ЗК* – затраты на хозяйственно-операционные нужды (бумага, литература, носители информации и т.п.), руб.;

Размер фонда оплаты труда разработчиков (*ЗФОТР*) рассчитывается по формуле:

(3.4)

где *ОРj* – месячный оклад j-го разработчика проектного решения, руб./мес.;

*ТРПРj* – время разработки проектного решения *j*-м разработчиком, мес. Включает в себя машинное время работы над проектом (*ТМРПР*);

*kД* – коэффициент дополнительной заработной платы разработчиков;

*kУ* – районный (уральский) коэффициент.

*ТРПР*=(515/(20,33\*8))/2=1,585 мес.

*ЗФОТР*=25000\*1,585\*(1+0,25)\*(1+0,15)= 56960,94 руб.

Так как разработкой занимаются 2 сотрудника

*ЗФОТР*= 56960,94 \* 2 = 113921,88 руб.

Величина отчислений во внебюджетные фонды определяется по формуле:

 (3.5)

где *kОВФ* – коэффициент отчислений во внебюджетные фонды.

Коэффициент отчислений во внебюджетные фонды рассчитывается по формуле:

kОВФ = kПФ + kФМС + kФСС + kСНС (3.6)

где *kПФ* – ставка отчислений в пенсионный фонд;

*kФМС* – ставка отчислений в фонд медицинского страхования;

*kФСС* – ставка отчислений в фонд социального страхования;

*kСНС* – ставка страховых взносов на обязательное социальное страхование от несчастных случаев на производстве и профессиональных заболеваний.

*kПФ = 22%*

*kФМС* *= 2,9%*

*kФСС* *= 5,1%*

*kСНС = 0,2%*

*ЗОВФ*=113921,88 \* 0,302= 34404,4 руб.

Таблица 3.11 – Затраты, связанные с эксплуатацией ЭВМ, определяются по формуле:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Вид оборудования | Мощность, кВт | Стоимость, 1 кВт/час | Время работы обор. Тож час | Сумма затрат, руб. |
| Компьютер | 0,40+0,40 | 4,5 | 379 | 1364,4 |
| Итого |  |  |  | 1364,4 |

*ЗЭВМ* = (М \* С) \* Т (3.7)

где М - Мощность, кВт;

С - Стоимость, 1 кВт/час;

Т - Время работы оборудования, час.

Высчитываем с учетов двух компьютеров разработчиков

*ЗЭВМ*=(0,4\*2\*4,5)\*379= 1364,4 руб.

Величина затрат на хозяйственно-операционные нужды на основании исходных данных определяется по формуле:

, (3.8)

где *Цτ*– цена *τ*-го товара, руб.;

*Qτ* – количество τ-го товара.

*ЗК*=150+2\*20=190 руб.

Результаты расчета представлены в таблице 3.12.

Таблица 3.12 – Затраты на разработку ИС

| Статья затрат | Сумма, руб. |
| --- | --- |
| Общий фонд оплаты труда разработчиков | 113921,88 |
| Отчисления во внебюджетные фонды с заработной платы разработчиков | 34404,4 |
| Затраты, связанные с эксплуатацией техники | 1364,4 |
| Затраты на хозяйственно-операционные нужды | 190 |
| Итого | 149880,68 |

Получаем, что *КРПР =* 149880,68 руб.

Итого получаем К=*КРПР + КОБ =* 149880,68 +211146,00 = 361026,68 руб.

## 3.3 Расчет текущих затрат до и после внедрения АИС.

Расчет текущих затрат до внедрения АИС

Текущие эксплуатационные затраты до внедрения ИС рассчитываются аналогично:

 (3.9)

где *ЗП1*– затраты на оплату труда специалистов до внедрения ИС, руб.;

*ОТвн1* – отчисления во внебюджетные фонды, руб.;

*ЗЭВМ1* – эксплуатационные затраты на ЭВМ, руб.;

Годовые затраты по заработной плате специалистов до внедрения АИС определяются по формуле:

ЗП1 = Ос\*Чс\*12\*(1+Ку)\*(1+Кд) (310)

где *Ос* – оклад специалиста, руб./мес.;

*Чс* – численность специалистов, участвующих в процессе, чел.;

Ку – региональный коэффициент;

Кд – коэффициент дополнительной заработной платы.

Затраты на заработную плату составят:

для диспетчера (1 человек):

*ЗП1 дисп* = (14500\*12) \*(1+0,25)\*(1+0,15)= 250125 руб.

*ЗЭВМ1 дисп* = 140\*(0,4\*4,5)=252 руб

для инженера ( 10 человек ):

*ЗП1 инж* = (20000\*12\*10)\* (1+0,25)\*(1+0,15)= 3450000 руб.

*ЗЭВМ1 инж* = 8,75\*(0,4\*10\*4,5)= 157,5 руб

общая сумма:

*ЗП1 = ЗП1 дисп + ЗП1 инж*

*ЗП1*= 250125 + 3450000 = 3700125 руб.

*ОТвн1*=3700125\* 0,302=1117437,75 руб.

*С*1 = *ЗП1 + ОТвн1 + ЗЭВМ1 инж + ЗЭВМ1 дисп*

*С*1=3700125 + 1117437,75 + 252 + 157,5 = 4817972,25 руб.

Расчет текущих затрат после внедрения АИС

После внедрения АИС на предприятии будет сокращен штат инженеров на 3 человек.

Текущие эксплуатационные затраты после внедрения ИС рассчитываются аналогично:

 (3.11)

где *ЗП2*– затраты на оплату труда специалистов до внедрения ИС, руб.;

*ОТвн2* – отчисления во внебюджетные фонды, руб.;

*ЗЭВМ2* – эксплуатационные затраты на ЭВМ, руб.;

ЗП2 = Ос\*Чс\*12\*(1+Ку)\*(1+Кд) (3.12)

где *Ос* – оклад специалиста, руб./мес.;

*Чс* – численность специалистов, участвующих в процессе, чел.;

Ку – региональный коэффициент;

Кд – коэффициент дополнительной заработной платы.

Затраты на заработную плату составят:

для диспетчера (1 человек):

*tобщ2 дисп*=1050/60=17,5 час.

*ЗП2 дисп*=(14500\*12) \*(1+0,25)\*(1+0,15)= 250125 руб.

*ЗЭВМ2 дисп*=17,5\*(0,4+4,5)=31,5 руб

для инженера (7 человек):

*tобщ2 инж*=105/60= 1,75 час.

*ЗП2 инж*= (20000\*12\*7)\* (1+0,25)\*(1+0,15)= 2415000 руб.

*ЗЭВМ2 инж*=1,75\*(0,4\*7\*4,5)= 22,05 руб

общая сумма:

*ЗП2 = ЗП2 дисп + ЗП2 инж*

*ЗП2*=250125 + 2415000 = 2665125 руб.

*ОТвн2*=2665125\*0,302= 804867,75 руб.

*С*2 = *ЗП2 + ОТвн2 + ЗЭВМ2 инж + ЗЭВМ2 дисп*

*С*2=2665125+ 804867,75 + (31,5 + 22,05) = 3470046,3 руб.

## 3.4 Расчет результатов от создания и использования АИС.

В качестве основного статического показателя экономической эффективности ЭИС может быть использован годовой экономический эффект (экономическая прибыль):

Ожидаемая экономия от внедрения системы рассчитывается по формуле:

 , (3.13)

где *Эож* – ожидаемая экономия от внедрения системы, руб.;

*С*1 – годовые текущие затраты до внедрения автоматизированной системы, руб.;

*С*2 – годовые текущие затраты после внедрения системы, руб.;

*Эож*=4817972,25 – 3470046,3 = 1347925,95 руб.

## 3.5 Расчет основных показателей экономической эффективности.

Величина срока окупаемости инвестиций определяется по формуле:

, (3.14)

Δ*Кt* – величина капитальных вложений, руб.;

Δ*Эt* – величина текущего эффекта в году, соответствующему полной компенсации капитальных затрат, руб.

Срок окупаемости составит:



Дисконтированный срок окупаемости, разработанной ИС составляет 0,3 года или 4 месяца, что меньше нормативного, следовательно, вложения выгодны.

Коэффициент эффективности капитальных затрат рассчитывается по формуле:

 (3.15)

Е=3,3

На один рубль капитальных вложений будет приходится 3,3 руб. ежегодной прибыли. Расчетный коэффициент больше нормативного, следовательно, вложения выгодны.

## 3.6 Результаты расчета.

Обобщенные результаты расчета экономической эффективности от внедрения проектируемой информационной системы учета движения производственных запасов представлены в таблице 3.13.

Таблица 3.13 – Показатели экономической целесообразности разработки и внедрения программного продукта

| Наименование показателя | До внедрения | После внедрения | Отклонения | |
| --- | --- | --- | --- | --- |
| +,- | % |
| Капитальные затраты, руб. | - | 361026,68 | + 361026,68 | - |
| Затраты на оплату труда специалистов, руб. | 3700125,00 | 2665125,00 | - 1035000,00 | 72,02 |
| Отчисления во внебюджетные фонды, руб. | 1117437,75 | 804867,75 | - 312570,00 | 72,02 |
| Экономия, руб. | - | 1347925,95 | - | - |
| Срок окупаемости, год | - | 0,3 | - | - |
| Коэффициент эффективности капитальных затрат | - | 3,3 | - | - |

Произведенные расчеты свидетельствуют, что внедрение разработанной информационной системы экономических расчетов по реализации продукции позволит сократить временные затраты на обработку заявок, что приведет к сокращению текущих затрат. До внедрения ИС годовые текущие затраты составляют 3700125 руб., после внедрения будут равны 2665125 руб., таким образом экономия составит 1347925,95 руб.

Срок окупаемости внедряемой системы меньше срока предположительного использования.

Опираясь на оценку экономической эффективности можно сделать вывод о том, что разработка и внедрение предлагаемой информационной системы является экономически обоснованной и целесообразной.



# 4 Информационная технология безопасности.

## 4.1 Информационная безопасность.

Для организации информационной безопасности и предотвращения несанкционированного доступа в разработанной АИС была организована авторизация пользователей по логину и паролю через соответствующую форму

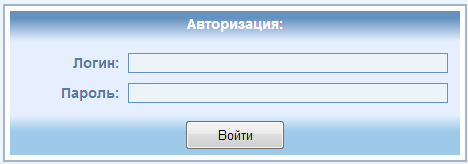


Рисунок 4.1 – Форма авторизации.

Для шифрование пароля был применен алгоритм хеширования MD5

Message Digest 5 (сокр. MD5) — результат трудов американского профессора информатики Рональда Ривеста из Массачусетского технологического Института и последний из серии разрабатываемых им криптографических алгоритмов по построению дайджестов сообщений. MD5. Первая публикация алгоритма MD5 появилась в 1991 г.

MD5 представляет собой хеш-функцию, которая вычисляет хэш (дайджест) для сообщения произвольной длины, подаваемого на её вход.

Хеш MD5 содержит 128 бит (16 байт) и, как правило, представляется в виде последовательности из 32 шестнадцатеричных чисел (по одной на четыре бита хеша). Хеш всегда имеет фиксированную длину.

Благодаря лавинному эффекту, изменение входного сообщения даже на один бит или символ приводит к полному изменению хеша. Сравните:

MD5("message digest") = f96b697d7cb7938d525a2f31aaf161d0

MD5("message digest.") = 6e823878de441602e3d095c956aefba3

Благодаря свойству необратимости хеширования, однозначно восстановить из хэша исходное сообщение невозможно, поскольку одному хэшу могут соответствовать множество различных сообщений. Именно поэтому хеш-функции называют односторонними.

Пароли пользователей хранятся в базе в виде MD5 хэшей. Вследствие чего даже получив прямой доступ к базе, пароли не буду скомпрометированы.

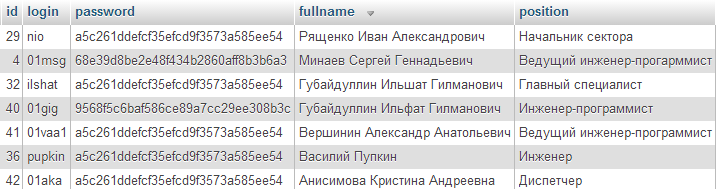


Рисунок 4.2 – Пароли пользователей в базе данных.

Так как существуют ресурсы, расшифровывающие MD5 хэши методом перебора по списку наиболее популярных паролей был применен способ изменения введенного пароля пользователя путем добавления к нему с конца некоторого набора символов. И уже данное сочетание шифруется и записывается в виде хэша в базу.

Приведем пример:

Пароль пользователя – qwerty

Добавочная часть пароля - hwu#{01Kvl

Добавочная часть имеет высокую сложность пароля. В нем присутствуют прописные буквы, строчные буквы, цифры, специальные символы, длина добавочной части 10 символов

Соединив пароль пользователя и добавочной части получаем итоговый пароль который шифруется:

qwertyhwu#{01Kvl

Полученный вариант пароля гарантирует, что даже если пользователь выбрал пароль, не отвечающий требования безопасности в базу данных записывается хэш надежного пароля.

Часть программного кода проверявшего соответствие пароля и логина:

<?php

//Начинаем сессию

session\_start();

//Подключаем конфигурационный файл

include("connect.php"); //Коннект до базы

$userstable = "hd\_user"; //база для коннекта

if (($\_POST[login]!='') and ($\_POST[password]!='')) //условие - если поля не пустые, то проверяем соответствие пароля и пользователя в базе данных

{

$query = "SELECT \* FROM $userstable WHERE login='".$\_POST[login]."'"; //запрос в базу

$query = mysql\_query($query) or die(mysql\_error());

$row=mysql\_fetch\_array($query);

$num = mysql\_num\_rows($query);

if ($num==1) //если соответствие пароля и пользователя найдено

{

if ($row['local']=="1" and $row['password']==md5($\_POST[password].$secretpass)) //шифруем пароль

{

$\_SESSION['login\_error']="1";

$\_SESSION['user\_login\_id']=$row[id];

$\_SESSION['fullname']=$row[fullname];

$\_SESSION['position']=$row[position];

$\_SESSION['access\_disp']=$row[access\_disp];

$\_SESSION['access\_ingener']=$row[access\_ingener];

$\_SESSION['access\_admin']=$row[access\_admin];

}

else {$\_SESSION['login\_error']="2";}

else //иначе если поля пустые то login\_error принимает значение Поля пустые

{

//echo "<meta http-equiv='refresh' content='0; url=index.php'>"; //перегружает страницу

$\_SESSION['login\_error']="3";

}

## 4.2 Обеспечение безопасности web-сервера, сервера баз данных и пользовательских компьютеров.

Обеспечение безопасности сервера баз данных и web-сервера являются основной, так как они являются центральным звеном хранение всей информации.

Методы защиты серверного оборудования на предприятии ОАО «Салаватнефтехимремстрой»:

* Блокирование доступа к серверу из глобальной сети интернет с помощью аппаратного шлюза безопасности.

На предприятии для этих целей используется шлюз безопасности Check Point, являющимся мировым лидером в данной области.

* Ограничение доступа в серверное помещение по электронным пропускам.

Ограничен доступ в серверное помещение только администраторам сервером по специальным электронным пропускам с фиксированием времени входа-выхода.

* Видеонаблюдение серверного помещения

Организованно видеонаблюдение серверного помещения с сроком хранения записей 3 месяца.

* Доступ к консоли сервера по протоколу SSH

Доступ к серверу имеют только администраторы по протоколу SSH. Пароли имеют необходимую сложность и своевременно обновляются.

Обеспечение безопасности пользовательских компьютеров осуществляется благодаря методу доступа к АИС с помощью авторизации по логину и паролю, а также с помощью антивирусного ПО.

На предприятии для антивирусной защиты используется Kaspersky Endpoint Security как на серверных, так и пользовательских операционных системах.

## 4.3 Электромагнитная безопасность.

Методы обеспечения электромагнитной безопасности:

* не размещать рабочие места с ПЭВМ вблизи источников электромагнитных полей (трансформаторов, мощных электропотребителей, распределительных щитов, кабельных подводов, радиопередающих устройств и др. источников ЭМП);
* перед установкой компьютерной техники обследовать помещение на наличие и интенсивность ЭМП промышленной частоты;
* заземлять массивные металлические элементы оборудования помещения (станки, стенды, оконные решетки и т.п.);
* размещать групповые рабочие места на нижних этажах зданий;
* заземлять все элементы оборудования ПЭВМ, если заземление оборудования ПЭВМ осуществляется через посредство третьего заземляющего проводника сети питания - проверить наличие и качество заземления путем замера сопротивления контура заземления;
* размещать провода питания, по возможности, в экранирующих металлических оболочках или трубах;
* оборудовать, по возможности, места группового подключения ПЭВМ (2 и более пользователей) экранированными щитками питания с необходимым количеством розеток;
* не использовать удлинители (переноски) и сетевые фильтры, выполненные в виде переносок.

Размещение компьютерной техники на рабочем месте.

* каждое рабочее место должно быть автономным;
* размещать экран дисплея на расстоянии не менее 50 см от глаз пользователя;
* размещать дисплей и системный блок ПЭВМ на максимально возможном расстоянии от пользователя;

Контроль электромагнитных полей компьютерной техники

Инструментальные измерения уровней ЭМП на рабочих местах выполняются:

* при вводе ПЭВМ в эксплуатацию, организации новых и реорганизации существующих рабочих мест;
* при аттестации рабочих мест по условиям труда;
* в порядке текущего надзора по заявкам организаций.

В соответствии с требованиями СанПин 2.2.2/2.4.1340-03 электромагнитная обстановка на рабочих местах оценивается по пяти параметрам:

* напряженности электрической составляющей ЭМП в диапазоне частот 5 Гц-2 кГц;
* напряженности электрической составляющей ЭМП в диапазоне частот 2 - 400 кГц;
* напряженности магнитной составляющей ЭМП в диапазоне частот 5 Гц-2 кГц;
* напряженности магнитной составляющей ЭМП в диапазоне частот 2 - 400 кГц;
* потенциалу на поверхности экрана дисплея (для дисплеев на основе ЭЛТ).

# Заключение

Часто стремительный рост требований и ожиданий бизнеса требует решительного внедрения требуемых мэр. Многие компании адекватно оценивают сложности, связанные с постановкой задачи, риски, связанные с требуемыми организационными изменениями, выбором и внедрением подходящего средства автоматизации, а также преимущества использования проверенного опыта. Однако ограничения бюджета могут быть «камнем преткновения» в привлечении внешней помощи.

Внедрение системы HelpDesk повышает эффективность работы ИТ отдела предприятия, снижает время реакции на проблему, позволяет минимизировать простой сотрудников предприятия из-за технических и программных проблем компьютерной и оргтехники, тем самым возрастает работоспособность предприятия в целом.

За время прохождения практики была обследованы входные и выходные данные, построены схема движения данных, спроектирована база данных. Было написан код рабочего места «Инженер», разработан интерфейс.

АИС «HelpDeskIT» находится на стадии разработки и тестирования, по окончании которой будет внедрена.

# Список литературы

1. Мишель Е.Д., Изучаем PHP и MySQL, 2-е издание, Символ-Плюс, 2008, 448 стр.
2. Овицкий, Н.Ф. PHP как инструмент для работы с MySQL базой данных, Москва, 2008.
3. Морозов, Б.С. MySQL для начинающих / Б.С. Морозов - СПб. : Корона-принт, 2007.
4. Демидов, С.Д. Технологии разработки WEB приложений / С.Д. Демидов, Н.А. Ерохин - СПб.: БХВ-Петербург, 2004.
5. Колесниченко, Д.С. CSS для начинающих WEB разработчиков / Д.С. Колесниченко - М. : ДНК, 2000.
6. Федорчук, Д. А. Разработка WEB приложений на PHP и MySQL / Д.А. Федорчук - СПб. : Корона-принт, 2003.
7. Прохоров, В.В. Разработка WEB приложений с нуля / В.В. Прохоров - Мн.: БГУИР, 2009.
8. Семич, А.В. Охрана труда при работе на персональных электронно-вычислительных машинах и другой офисной технике. Практическое пособие / А.В. Семич – Мн.: ЦОТЖ, 2001.
9. Сорокин, А.В. Приёмы работы с MySQL базой данных / А.В. Сорокин – М.: 2003.
10. Федоров, Н.З. PHP и MySQL для начинающих / Н.З. Федоров - СПб.: БХВ-Петербург, 2001.
11. Мариков, Ф.Ф. Самоучитель PHP 5 / Ф.Ф Мариков - М. : Вильямс, 2001.
12. Программирование на платформе Microsoft .NET ISBN 5-7502-0088-4, 5-469-00820-7, 0-7356-1422-9; 2005 г.