

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное автономное образовательное учреждение
высшего образования

«ЮЖНО-УРАЛЬСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
(НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ)»

Факультет электротехнический

Кафедра автоматики

Направление подготовки 27.03.04 «Управление в технических системах»

ДОПУСТИТЬ К ЗАЩИТЕ

Заведующий кафедрой

_____ Голощапов С.С. _____

_____ 2020 г.

Программное обеспечение технической поддержки пользователей
информационных сервисов

ПОЯСНИТЕЛЬНАЯ ЗАПИСКА
К ВЫПУСКНОЙ КВАЛИФИКАЦИОННОЙ РАБОТЕ
ЮУрГУ–27.03.04.2020.046.00.00 ПЗ ВКР

Автор ВКР

студент группы МиЭт-527

_____ / Вагнер К.А. _____
подпись ФИО

_____ 2020 г.

Руководитель работы

К. Т. Н., доцент

_____ должность

_____ / Голощапов С.С. _____
подпись ФИО

_____ 2020 г.

Консультант

_____ должность

_____ / Данилов Н.Г. _____
подпись ФИО

_____ 2020 г.

Нормоконтроль

старший преподаватель

_____ должность

_____ / Елисеев В.П. _____
подпись ФИО

_____ 2020 г.

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное автономное образовательное учреждение
высшего образования

«ЮЖНО-УРАЛЬСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
(НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ)»

Факультет электротехнический

Кафедра автоматики

Направление подготовки 27.03.04 «Управление в технических системах»

УТВЕРЖДАЮ
Заведующий кафедрой

Голощапов С.С.

2020 г.

ЗАДАНИЕ

на выпускную квалификационную работу студента

Вагнера Константина Алексеевича

(Ф. И.О. полностью)

Группа МиЭт-527

1 Тема работы «Программное обеспечение технической поддержки пользователей информационных сервисов»

утверждена приказом по университету от __. __.2020 г. № __

2 Срок сдачи студентом законченной работы: 11.06.2020 г.

3 Исходные данные к работе

1. Техническое задание на разработку программного обеспечения.

2. Наличие ПЭВМ, подключенной к компьютерной сети предприятия. Для работы с системой достаточно квалификации пользователя ПК.

3. Функционирование системы осуществляется в среде операционной системы Microsoft Windows 7, 10. Язык разработки – PHP 7.4.1. СУБД – ORACLE 9.

Основные требования к ВКР: разработать программное обеспечение, основной задачей которого является накопление базы знаний, координация работ по устранению сбоев в ИТ-инфраструктуре, а также по предоставлению и обслуживанию доступа к ИТ-услугам.

Программное обеспечение должно включать в себя ряд модулей, обеспечивающих выполнение следующих функций: управление проблемами, управление изменениями, управление конфигурациями и конфигурационными единицами.

Программное обеспечение должно быть разработано на языке разработки PHP 7.4.1 и обладать возможностью расширения функциональных возможностей.

6 Консультанты по работе, с указанием относящихся к ним разделов работы

Раздел	Консультант	Подпись, дата	
		Задание выдал (консультант)	Задание принял (студент)

7 Дата выдачи задания 03 февраля 2020 года

Руководитель _____ / С.С.Голощанов
(подпись) (И.О. Ф.)

Консультант _____ / Н.Г.Данилов
(подпись) (И.О. Ф.)

Задание принял к исполнению _____ / К.А. Вагнер
(подпись студента) (И.О. Ф.)

КАЛЕНДАРНЫЙ ПЛАН

Наименование этапов выпускной квалификационной работы	Срок выполнения этапов работы	Отметка о выполнении руководителя
1. Анализ технического задания (исходных данных) и исследование алгоритма работы системы.	03.02.2020 – 17.02.2020	
2. Изучение технологии Service Desk, методологии ITIL/ITSM.	18.02.2020 – 15.03.2020	
3. Выбор инструментальных средств создания программного обеспечения.	16.03.2020 – 20.03.2020	
4. Разработка схемы базы данных.	21.03.2020 – 24.03.2020	
5. Изучение языка программирования PHP.	25.03.2020 – 20.04.2020	
6. Разработка алгоритмов программного обеспечения.	21.04.2020 – 25.04.2020	
7. Написание программного кода.	26.04.2020 – 16.05.2020	
8. Отладка программного обеспечения.	17.05.2020 – 31.05.2020	
9. Оформление пояснительной записки.	01.06.2020 – 11.06.2020	

Заведующий кафедрой _____ /С.С. Голощапов /
(подпись) (И.О. Ф.)

Руководитель работы _____ /С.С. Голощапов /
(подпись) (И.О. Ф.)

Консультант _____ / Н.Г. Данилов /
(подпись) (И.О. Ф.)

Студент _____ /К.А. Вагнер /
(подпись) (И.О. Ф.)

АННОТАЦИЯ

Вагнер К.А. Выпускная квалификационная работа «Программное обеспечение технической поддержки пользователей информационных сервисов» – Миасс: ЮУрГУ – 2020 г. – 80 с., 9 ил., 18 табл., библиогр. список – 19 наим., 1 прил.

В ходе написания работы было разработано программное обеспечение технической поддержки пользователей информационных сервисов, программный код реализован на языке PHP.

Программное обеспечение реализует следующие функции:

- все сведения о сбоях, изменениях регистрируются и хранятся в единой базе данных;
- имеется возможность для пользователей контролировать ход восстановительных работ, получая автоматизированные почтовые уведомления о смене статуса их запроса и предпринятых действиях;
- имеется доступ для специалистов к постоянно накапливающимся сведениям о произошедших сбоях, проведенных изменениях в привязке к определенной конфигурационной единице, а также известным ошибкам.
- отслеживание эффективной работы ИТ-специалистов в режиме реального времени на основе web-технологий.

					<i>27.03.04.2020.046.00.00 ПЗ ВКР</i>			
<i>Изм.</i>	<i>Лист</i>	<i>№ докум.</i>	<i>Подпись</i>	<i>Дата</i>				
<i>Разраб.</i>		<i>Вагнер К.А.</i>			<i>Программное обеспечение технической поддержки пользователей информационных сервисов</i>	<i>Лит</i>	<i>Лист</i>	<i>Листов</i>
<i>Провер.</i>		<i>Голощанов С.С.</i>				<i>К</i>	<i>6</i>	<i>97</i>
<i>Н. Контр.</i>		<i>Елисеев В.П.</i>				<i>ЮУрГУ Кафедра «Автоматика»</i>		
<i>Утверд.</i>		<i>Голощанов С.С.</i>						

ОГЛАВЛЕНИЕ

ВВЕДЕНИЕ	8
1 ИТ-ИНФРАСТРУКТУРА НА ПРЕДПРИЯТИИ.....	12
1.1 Понятие ИТ-инфраструктуры	12
1.2 Описание корпоративного web-портала	12
2 ОБОСНОВАНИЕ ТЕХНИЧЕСКОЙ ПРОБЛЕМЫ И МЕТОДЫ ЕЕ РЕШЕНИЯ	14
2.1 Обоснование технической проблемы.....	14
2.2 Методы решения технической задачи	14
3 ТЕХНОЛОГИЯ SERVICE DESK.....	19
4 ВЫБОР ИНСТРУМЕНТАЛЬНЫХ СРЕДСТВ СОЗДАНИЯ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ	20
5 РАЗРАБОТКА БЛОК-СХЕМ АЛГОРИТМОВ	40
6 ИНФОРМАЦИОННОЕ И ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ КОМПЛЕКСА ЗАДАЧ	45
ЗАКЛЮЧЕНИЕ	65
БИБЛИОГРАФИЧЕСКИЙ СПИСОК	66
ПРИЛОЖЕНИЕ А	68

					<i>27.03.04.2020.046.00.00 ПЗ ВКР</i>	<i>Лист</i>
<i>Изм.</i>	<i>Лист</i>	<i>№ докум.</i>	<i>Подпись</i>	<i>Дата</i>		<i>7</i>

ВВЕДЕНИЕ

На сегодняшний день успех компаний все больше зависит от используемых информационных технологий (далее – ИТ). Рост зависимости ведет к росту требований к качеству предоставляемых ИТ-услуг.

ИТ, создававшиеся для того, чтобы обслуживать бизнес, сами нуждаются в управлении. И чем больше размер компании, количество пользователей и активно используемых элементов ИТ-инфраструктуры, тем острее проблема менеджмента этих ресурсов.

ИТ-поддержка — это обслуживание и модернизация ИТ-инфраструктуры на предприятии. Поддержка выражается в профессиональном управлении оборудованием и ПО, которые используются для правильного протекания бизнес-процессов.

Новые компании нуждаются в оперативной настройке компьютерного оборудования и стабильной работе ИТ-отдела. Малый бизнес не всегда может выделить одну или несколько штатных единиц на развитие и поддержание ИТ-инфраструктуры. У крупного бизнеса возникают проблемы с ростом объема данных и обслуживанием удаленных офисов и филиалов.

Одним достаточно удаленной технической поддержки и консультаций, а также регулярного приезда опытного инженера. Другим необходим постоянный контроль и участие в работе ИТ-инфраструктуры.

Что входит в ИТ-поддержку

- Проведение ИТ-аудита, анализ текущей ситуации.
- Мероприятия для стабильной работы оборудования и ПО.
- Администрирование серверной группировки, компьютерной сети и телефонии.
- Модернизация оборудования, апгрейд ПО.
- Обслуживание пользовательских компьютеров.

					<i>27.03.04.2020.046.00.00 ПЗ ВКР</i>	Лист
Изм.	Лист	№ докум.	Подпись	Дата		8

- Ликвидация последствий вирусной атаки, настройка антивирусного ПО.
- Мероприятия по повышению корпоративной безопасности.
- Масштабирование инфраструктуры под новые потребности бизнеса.
- Развертывание и настройка новых продуктов и сервисов.
- Ремонт оборудования, включая периферийное и сетевое.
- Организация переезда офиса (только ИТ-составляющая).
- Консультации пользователей по техническим вопросам.
- Работа сисадмина на объекте (для устранения неполадок и других задач).

Служба технической поддержки на каждом предприятии может быть построена разнообразными способами (имеется в виду реализации процессов поддержки). Существует несколько моделей службы поддержки, например: централизованная, локальная, виртуальная — с единым телефонным центром и т. д. Служба технической поддержки может быть организована как в целях обслужить внешних клиентов (аутсорсинг обслуживания компьютеров и т. п.), так и внутренних (подразделение ИТ-департамента на крупных предприятиях).

В описании концепции ITIL, построенной на процессном подходе, Service Desk является единой точкой контакта для пользователей ИТ-Услуг. Это исключение сделано ввиду большой важности подразделения техподдержки и при внедрении практическом использовании современных ИТ-подходов и методик.

Правильно организованная техподдержка (Service Desk) всегда начинается с регистрации всех обращений конечных пользователей, служит единой точкой для общения пользователя с ИТ-службой. Наиболее популярные решения по практической организации техподдержки часто строятся на базе Call-center. Он является начальной точкой контактов конечных пользователей со службой техподдержки и служит источником

					<i>27.03.04.2020.046.00.00 ПЗ ВКР</i>	Лист
Изм.	Лист	№ докум.	Подпись	Дата		9

информации об их фактической удовлетворенности уровнем сервиса, что дополняет информацию о технических параметрах качества обслуживания компании-клиента (внешнего или внутреннего).

На больших предприятиях или в крупных компаниях-аутсорсерах служба технической поддержки часто организована по следующему многоуровневому принципу:

- Пользователь — обращается с вопросом в службу поддержки по телефону или с помощью электронной заявки (электронная почта, или специальные сервисы подачи заявок).
- Оператор (1-я линия поддержки, Call-center) — регистрирует обращение, при возможности помогает пользователю самостоятельно, либо эскалирует (передает и контролирует выполнение) заявку на вторую линию поддержки.
- Вторая линия поддержки — получает заявки от первой линии, работает по ним, при необходимости привлекая к решению проблемы специалистов из смежных отделов (например, системные администраторы, поддержка POS-терминалов, поддержка специального ПО, поддержка специального оборудования, администраторы биллинговой системы и т. д.).

По сути система технической поддержки или техподдержка (Technical support, Helpdesk, Service desk) — сервисная структура, разрешающая проблемы пользователей с компьютерами, аппаратным и программным обеспечением. Важная функциональная составляющая ИТІЛ (IT Infrastructure Library, библиотека инфраструктуры ИТ), позволяющая выявить проблемные участки инфраструктуры ИТ, оценить эффективность работы подразделения ИТ.

Первым шагом на пути реформирования ИТ-процессов в соответствии с ИТІЛ является создание службы Service Desk. Внедрение этой системы позволяет автоматизировать процесс сопровождения предоставляемых заказчику ИТ-услуг. Такая система незаменима в компаниях, чьи подразделения значительно распределены территориально и имеют

					<i>27.03.04.2020.046.00.00 ПЗ ВКР</i>	Лист
						10
Изм.	Лист	№ докум.	Подпись	Дата		

разветвленную, а, следовательно, непростую с точки зрения управления, ИТ-инфраструктуру.

Service Desk предназначен для обработки инцидентов, проблем, а также обеспечивает интерфейс для других видов деятельности, таких как запросы на изменения, управление конфигурациями. Доступность оперативных данных, использование сохраненного опыта, наличие контроля со стороны конечного пользователя, все это приводит к улучшению качества предоставляемых услуг.

Еще одно ключевое преимущество Service Desk состоит в обеспечении информацией для принятия решений, в частности, данными о состоянии ИТ-инфраструктуры, качестве предоставляемых услуг и эффективности использования персонала. Эта информация представляет значительную ценность не только для ИТ-менеджеров, но и для руководителей бизнес-подразделений предприятия, которым трудно оценить эффективность вложений в ИТ.

					<i>27.03.04.2020.046.00.00 ПЗ ВКР</i>	<i>Лист</i>
<i>Изм.</i>	<i>Лист</i>	<i>№ докум.</i>	<i>Подпись</i>	<i>Дата</i>		<i>11</i>

1 ИТ-ИНФРАСТРУКТУРА НА ПРЕДПРИЯТИИ

1.1 Понятие ИТ-инфраструктуры

Понятие ИТ-инфраструктуры включает в себя большое количество персональных компьютеров и принтеров, сложное сетевое оборудование, серверы, а также разнообразие программного обеспечения и периферии.

В связи с этим возникают следующие проблемы:

- как управлять таким объемом инфраструктуры;
- как наиболее просто и с минимальными затратами времени использовать информационные ресурсы на предприятии, а также осуществлять их поиск и обработку с любого рабочего места.

1.2 Описание корпоративного web-портала

Портальные технологии позволяют объединить территориально распределенные части инфраструктуры предприятия и обеспечить единообразный доступ пользователей к информационным ресурсам.

Цель корпоративного информационного портала – предоставить работникам предприятия возможность из единой «унифицированной» точки получать доступ ко всем источникам данных предприятия.

Преимущества web-портала:

- возможность доступа ко всем ресурсам – рабочее место работника;
- доступ к корпоративной информационной системе через стандартный web-обозреватель – Internet Explorer, Opera и т.д.;
- единое представление информационных ресурсов через web-интерфейс;
- единая технологическая платформа для работы функциональных приложений;

					27.03.04.2020.046.00.00 ПЗ ВКР	Лист
Изм.	Лист	№ докум.	Подпись	Дата		12

- интеграция бизнес-процессов и приложений;
- функции распределенной работы;
- дополнительные информационные сервисы.

Архитектура web-сервисов показана на рисунке 1.1.



Рисунок 1.1 – Архитектура web-сервисов

2 ОБОСНОВАНИЕ ТЕХНИЧЕСКОЙ ПРОБЛЕМЫ И МЕТОДЫ ЕЕ РЕШЕНИЯ

2.1 Обоснование технической проблемы

В результате изменения ИТ-инфраструктуры на предприятии в работе ИТ-служб выявляется ряд проблем:

- нет структурированного механизма поддержки клиентов;
- время ответа и качество предоставляемой поддержки не адекватны требованиям пользователей;
- нет измеряемых характеристик качества работы группы поддержки;
- специалисты ИТ загружены неравномерно, служба поддержки находится в состоянии перманентного аврала;
- отсутствует системное решение вопросов накопления и повторного использования знаний по устранению инцидентов;
- имеют место нескоординированные и неформализованные изменения;
- нет доступной информации для менеджмента – для принятия обоснованного решения не хватает фактов.

Учитывая перечисленные выше проблемы, можно обозначить основные задачи:

- предотвращение возможных потерь пользовательских запросов;
- сокращение времени ответа на пользовательские запросы;
- организация базы знаний, автоматизация создания отчетности;
- координация деятельности по предоставлению ИТ-услуг.

2.2 Методы решения технической задачи

Методологической базой был выбран опыт библиотеки ITIL, а именно модели процессов ITSM.

					27.03.04.2020.046.00.00 ПЗ ВКР	Лист
Изм.	Лист	№ докум.	Подпись	Дата		14

В основе методологии – универсальный опыт организации ИТ-процессов в компаниях различного масштаба и направлений деятельности.

Соблюдение стандартов, изложенных в библиотеке, позволяет, используя процессный подход, организовать работу ИТ-подразделения более эффективно.

В модели ITSM выделяются две группы процессов — процессы сопровождения услуг и процессы предоставления услуг (рисунок 2.1) .



Рисунок 2.1 – Модель ITSM

ИТ-услуга – набор информационных приложений, которые могут использоваться бизнес-пользователями Компании. Примеры ИТ-услуг: электронная почта, функциональность ERP-системы, офисные приложения.

Первая группа процессов относится к операционному уровню и обеспечивает повседневную (то есть операционную) деятельность ИС (рисунок 2.2).

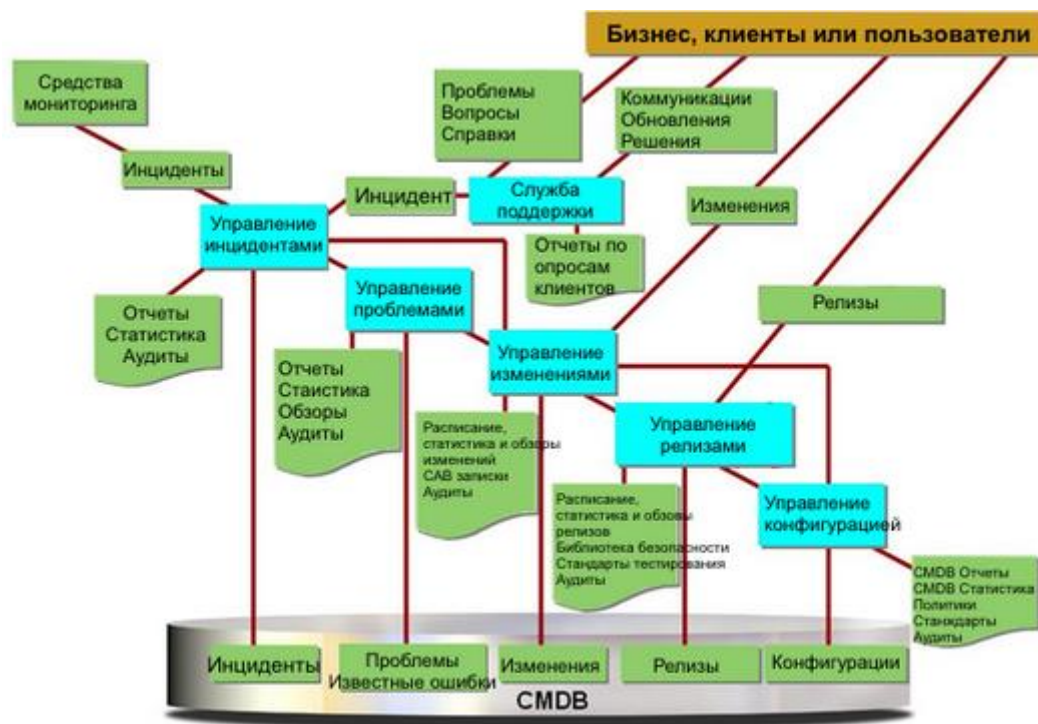


Рисунок 2.2 – Модель процессов управления услугами

Процессы предоставления услуг обеспечивают постановку целей, задач и метрик для процессов операционного уровня, они относятся к тактическому уровню.

Процессы сопровождения услуг осуществляют доступ пользователей к ИТ-услугам. В этой группе выделяются следующие процессы:

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

Управление инцидентами в модели ITIL/ITSM обеспечивает скорейшее восстановление ИТ-услуги в случае возникновения любого события, не являющегося частью нормального функционирования услуги, которое приводит или может привести к перерыву в обслуживании или снижению его качества.

Процесс управления инцидентами включает следующие операции:

- обнаружение и регистрация инцидента в базе данных;
- классификация и начальная поддержка;
- изучение и диагностика инцидента;
- разрешение инцидента и восстановление ИТ-услуги (выполняет владение инцидентом, его мониторинг, отслеживание и коммуникация - поддержание контакта с пользователем и специалистами службы ИС в ходе разрешения инцидента).

В рамках корректно построенного процесса управления инцидентами решается и задача диспетчирования работ службы ИС. Однако для этого требуется единая база данных работ, то есть единая база данных инцидентов. Соответственно, централизация запросов пользователей на Service Desk — необходимое условие эффективного управления инцидентами.

Управление проблемами в модели ITIL/ITSM обеспечивает выявление и устранение ошибок в инфраструктуре ИТ, что снижает количество инцидентов и повышает надежность инфраструктуры ИТ. Под проблемой в ITIL/ITSM понимается невыясненная корневая причина одного или нескольких инцидентов. Цель управления проблемами — установить корневую причину и найти обходной путь, или, в терминах модели, превратить проблему в известную ошибку. Известная ошибка затем разрешается посредством запроса на изменение и его реализации.

Управление конфигурациями в модели ITIL/ITSM решает следующие задачи:

- учет всех активов ИТ в организации;
- предоставление точной информации по оборудованию, ПО и документации на них для поддержки других процессов ITIL/ITSM;
- обеспечение выполнения процессов управления инцидентами, проблемами, изменениями и релизами;
- сравнение записей конфигурации с инфраструктурой ИТ и исправление ошибок.

					<i>27.03.04.2020.046.00.00 ПЗ ВКР</i>	<i>Лист</i>
<i>Изм.</i>	<i>Лист</i>	<i>№ докум.</i>	<i>Подпись</i>	<i>Дата</i>		<i>17</i>

Конфигурационная единица — это экземпляр оборудования, ПО, документации, связанный с инфраструктурой ИТ.

БД конфигурационных единиц обеспечивает достоверную информацию по инфраструктуре ИТ. Например, в процессе управления инцидентами БД позволяет установить находящееся у пользователя оборудование и ПО, не посещая рабочее место пользователя.

Управление изменениями в модели ITIL/ITSM обеспечивает применение стандартизованных процедур и методов для эффективной и быстрой обработки всех изменений для снижения их негативного влияния на качество ИТ-услуги. В данном процессе можно выделить три основных вида деятельности:

- регистрация запросов на изменения и фактически выполненных изменений (выполняется в процессе управления конфигурациями);
- авторизация изменений - разрешение или запрещение изменений, запрошенных бизнес-заказчиками;
- реализация изменений - действия по реализации изменений, согласованные друг с другом, а также с иными работами, выполняемыми ИТ-подразделением.

Управление релизами в модели ITIL/ITSM обеспечивает целостный взгляд на изменение, гарантирующий, что все аспекты релиза (технические и нетехнические) будут рассматриваться во взаимосвязи. Релиз — это набор новых и измененных элементов конфигурации, которые тестируются и внедряются совместно.

					<i>27.03.04.2020.046.00.00 ПЗ ВКР</i>	<i>Лист</i>
<i>Изм.</i>	<i>Лист</i>	<i>№ докум.</i>	<i>Подпись</i>	<i>Дата</i>		<i>18</i>

3 ТЕХНОЛОГИЯ SERVICE DESK

Чтобы совместить требования клиентов и бизнеса создается централизованная точка входа для обработки запросов клиентов. Эта единая точка может называться по-разному и выполнять функции разных структур.

Call Center. Основной акцент сделан на профессиональную обработку большого количества телефонных транзакций.

Help Desk. Основная функция - управление, координация и разрешение инцидентов максимально быстро и недопущение того, чтобы запросы терялись, забывались или игнорировались.

Service Desk. По сравнению с Help Desk диапазон услуг расширен, используются более глобальные подходы. Функции Service Desk не ограничиваются только обработкой инцидентов, проблем и вопросов, но также обеспечивают интерфейс для других видов деятельности, таких как запросы на изменения, управление конфигурациями, управление уровнем сервиса.

Ключевое преимущество ПО Service Desk состоит в обеспечении информацией для принятия решений, в частности, данными об использовании ресурсов персонала, недостатке услуг, производительности, выявлении необходимости обучения клиентов.

					<i>27.03.04.2020.046.00.00 ПЗ ВКР</i>	<i>Лист</i>
<i>Изм.</i>	<i>Лист</i>	<i>№ докум.</i>	<i>Подпись</i>	<i>Дата</i>		<i>19</i>

4 ВЫБОР ИНСТРУМЕНТАЛЬНЫХ СРЕДСТВ СОЗДАНИЯ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ

На сегодняшний день рынок предлагает большое количество программных продуктов, предназначенных для реализации Service Desk. Среди ведущих можно назвать CA Advanced Help Desk, HP OpenView Service Desk, Peregrine ServiceCenter, Remedy Action Request System, Tivoli Service Desk. Однако стоимость внедрения этих решений с учетом цены не только самого ПО, но и работы консультантов, расходов на обучение пользователей чрезвычайно велика.

Разрабатываемый проект реализуется на базе технологии «клиент-сервер». Снижение сетевого трафика при выполнении запросов (запрос выполняется непосредственно на сервере баз данных, на рабочую станцию передается только результат выполнения запроса, а не весь массив данных).

Возможность хранения бизнес-правил на сервере, что позволяет избежать дублирования кода в различных приложениях, использующих общую базу данных.

Возможность централизованного администрирования, управления безопасностью и правами доступа.

Клиент-серверная информационная система состоит из трех основных компонентов:

- сервер баз данных, управляющий хранением данных, доступом и защитой, резервным копированием, отслеживающий целостность данных в соответствии с бизнес-правилами и выполняющий запросы клиента;
- клиент, предоставляющий интерфейс пользователя, выполняющий логику приложения, проверяющий допустимость данных, посылающий запросы к серверу и получающий ответы от него;
- сеть и коммуникационное программное обеспечение, осуществляющее взаимодействие между клиентом и сервером посредством сетевых протоколов.

					27.03.04.2020.046.00.00 ПЗ ВКР	Лист
Изм.	Лист	№ докум.	Подпись	Дата		20

В качестве СУБД было задано ПО ORACLE9. СУБД обладает широкими возможностями управления пользовательскими привилегиями и правами доступа к различным объектам базы данных, резервного копирования и архивации данных, а также оптимизации выполнения запросов.

Это первая в мире автономная база данных, и она переопределяет управление базами данных с помощью машинного обучения и автоматизации для устранения человеческого труда, человеческих ошибок и ручной настройки, снижения затрат и сложности и обеспечения более высокой надежности, безопасности и большей операционной эффективности. Автономная база данных поддерживает сложное сочетание высокопроизводительных транзакций, включая отчеты, пакетные операции, Интернет вещей (IoT) и машинное обучение в единой базе данных. Это упрощает разработку и развертывание приложений, а также обеспечивает аналитику в реальном времени, персонализацию и обнаружение мошенничества.

Oracle Database Enterprise Edition - основа автономной базы данных, начинается с Oracle Database Enterprise Edition, которая является той же самой базой данных, которую клиенты использовали в течение многих лет для выполнения всех видов рабочих нагрузок, включая хранилище данных, аналитику и обработку транзакций. Автономная база данных использует автоматические параметры конфигурации и устраняет сложность управления, как описано далее в этом документе. Автономная база данных построена поверх и автоматизирует многие передовые технологии баз данных, уникальные для Oracle, включая: реальные кластеры приложений для масштабирования, отказоустойчивости, и онлайн-исправления; онлайн-операции для изменения схемы; активная защита данных для аварийного восстановления с учетом базы данных; параллельный SQL высокая пропускная способность; контейнеры баз данных для гибкости; Database In-Memory высокая производительность; прозрачное шифрование баз данных для защиты данных; хранилище баз данных для разделения ролей.

					<i>27.03.04.2020.046.00.00 ПЗ ВКР</i>	<i>Лист</i>
<i>Изм.</i>	<i>Лист</i>	<i>№ докум.</i>	<i>Подпись</i>	<i>Дата</i>		<i>21</i>

Oracle Cloud спроектирован для полного спектра приложений и баз данных, от небольших до приложений бизнес-класса, требующих высокой производительности и масштабируемости. У Oracle есть тысячи клиентов, работающих с критически важными системами, которые продолжают расширять границы вычислительных технологий. Клиенты Oracle имеют рабочие нагрузки, превышающие возможности самых больших компьютеров, доступных на рынке, и только база данных Oracle способна удовлетворить эти потребности с помощью своей технологии кластеризации-реальных кластеров приложений. Облачная инфраструктура Oracle разработана с учетом этих потребностей для обеспечения непревзойденной производительности, безопасности и доступности в облаке. Oracle Best Practices Oracle публикует рекомендации "best practice" по развертыванию, настройке и эксплуатации таких продуктов, как база данных Oracle и Exadata. Эти рекомендации "наилучшей практики" развиваются вместе с эволюцией базовых компонентов инфраструктуры и базы данных Oracle. Команда разработчиков баз данных Oracle-это ведущие специалисты в области технологий Oracle, и это та же команда, которая разработала автономную базу данных. Oracle работает с клиентами по всему миру, чтобы определить наилучшие методы использования технологии Oracle для достижения высочайшей доступности, безопасности и производительности, и внедрила те же самые лучшие методы в автономную базу данных. База знаний Oracle Knowledge Base Oracle выходит за рамки команды разработчиков Oracle и включает в себя базу данных Oracle internaldatabase запросов на обслуживание, запросов на улучшение и базы данных ошибок. Эти базы данных включают связанные файлы трассировки, файлы журналов и другую вспомогательную информацию, используемую для диагностики и устранения неполадок. Эта база знаний формирует входные данные, необходимые для построения, и обучает модели машинного обучения, необходимые для автоматизации решения проблем в Автономной базе данных. Размер баз данных машинного обучения может варьироваться от нескольких мегабайт до сотен терабайт и выше. Базы данных Oracle могут

					<i>27.03.04.2020.046.00.00 ПЗ ВКР</i>	<i>Лист</i>
<i>Изм.</i>	<i>Лист</i>	<i>№ докум.</i>	<i>Подпись</i>	<i>Дата</i>		22

состоять всего из нескольких объектов или миллиардов объектов. Рабочие нагрузки могут включать обслуживание веб-сайтов, финансовых систем, медицинских записей, производства, ERP или практически любого мыслимого бизнеса. Машинное обучение играет ключевую роль в автоматизации, определяющей автономную базу данных. Машинное обучение - это также набор инструментов, доступных в Oracle Cloud, которые клиенты могут использовать для реализации своих собственных решений.

Базовые конвергентные возможности базы данных Oracle Database позволяют предлагать автономную базу данных в двух изданиях, которые специально адаптированы к рабочей нагрузке в соответствии с рекомендациями Oracle Best Practice (рисунок 4.1).

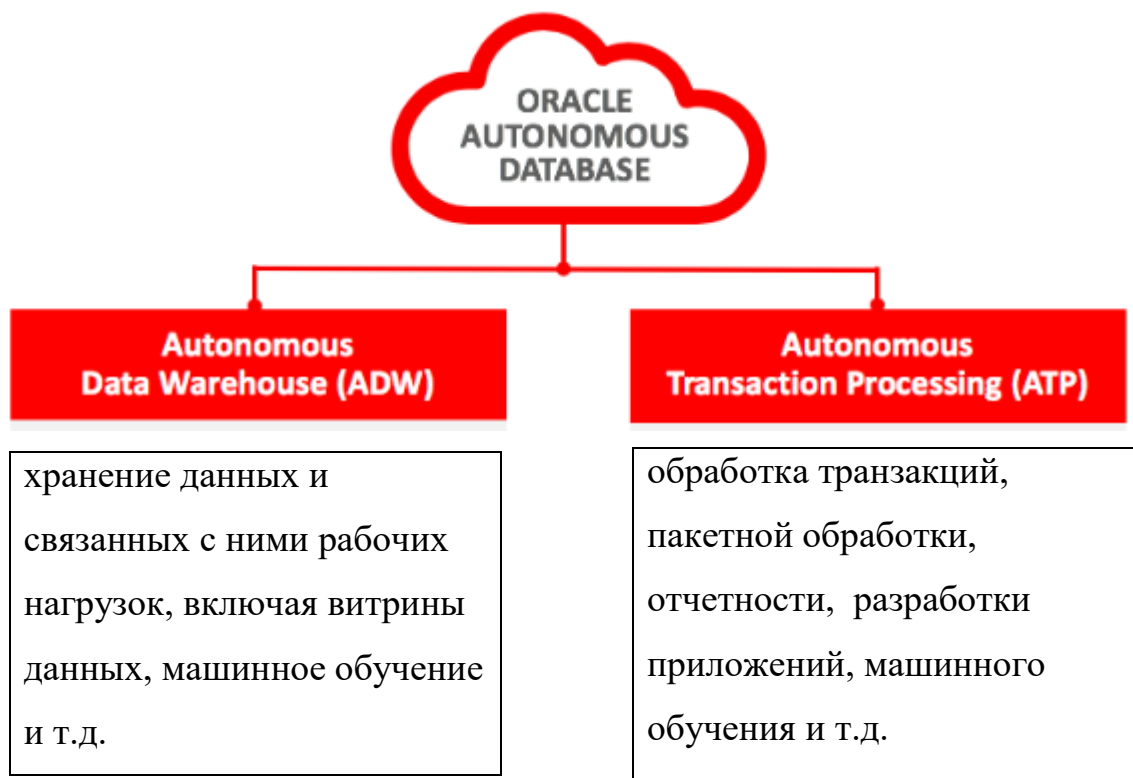


Рисунок 4.1 – Автономная база данных Oracle Database

Oracle Autonomous Transaction Processing (ATP) адаптирована для оперативной обработки транзакций, пакетной обработки, отчетности, Интернета, разработки приложений, машинного обучения и смешанных сред рабочей нагрузки.

Автономное хранилище данных как следует из названия, Oracle Autonomous Data Warehouse (ADW) адаптировано для хранения данных и связанных с ними рабочих нагрузок, включая витрины данных, машинное обучение или как часть развертывания озера данных. Эти системы и базы данных, как правило, отделены от приложений обработки транзакций и построены для удовлетворения конкретных бизнес-потребностей. Хранилища данных часто используют подходы моделирования данных, такие как Звездные схемы и другие методы, чтобы обеспечить соответствие структур данных потребностям бизнес-пользователей, проводящих анализ данных, и специалистов по анализу данных, выполняющих анализ тенденций. Хранилища данных обычно содержат большие объемы данных, которые обрабатываются в bulkog потоком в базу данных. Хранилища данных часто полагаются на представление сводных данных и сильно параллельный SQL, чтобы обеспечить быстрое время отклика. Автономное хранилище данных Oracle разработано специально для таких случаев использования. Автономная обработка транзакций Oracle Autonomous Transaction Processing (ATP) привносит те же автономные возможности, что и ADW, в область обработки транзакций и смешанной рабочей нагрузки. ATP ориентирована в первую очередь на сложную обработку транзакций, которая включает в себя оперативную отчетность и/или пакетную обработку данных. Возможность выполнения смешанных рабочих нагрузок в одной базе данных устраняет необходимость перемещения данных из базы данных обработки транзакций в отдельную систему отчетов или аналитики. Эта возможность снижает сложность приложений и устраняет время ожидания, связанное с перемещением данных между службами обработки транзакций и аналитическими базами данных. ATP поддерживает IoT и машинное обучение в дополнение к OLTP. Автоматизируя создание баз данных и управление ими, ATP также значительно упрощает разработку приложений.

Автономная база данных может быть развернута как на общей, так и на выделенной инфраструктуре Exadata. Что касается общей инфраструктуры,

					<i>27.03.04.2020.046.00.00 ПЗ ВКР</i>	Лист
Изм.	Лист	№ докум.	Подпись	Дата		24

Oracle берет на себя полную ответственность за все аспекты сервисного обслуживания. Все клиенты используют одну и ту же общую инфраструктуру Exadata, управляемую Oracle, с полной изоляцией данных и системных ресурсов для каждого клиента. Клиенты просто указывают, какие ресурсы (процессор и хранилище) им требуются и в каком регионе. Oracle автоматически берет на себя заботу о работе всех баз данных и систем. В общей инфраструктуре развертывания клиенты не участвуют в операционных задачах, таких как обновление программного обеспечения. Операции выполняются экспертами Oracle на основе лучших практик. Общая инфраструктура отлично подходит для клиентов, которые хотят быть пользователями баз данных, не беспокоясь о каких-либо операциях с базами данных. Вариант развертывания общей инфраструктуры требует минимальных обязательств всего в один час и один час. OCPU.It может быть мгновенно масштабирован как с точки зрения процессора, так и с точки зрения хранения данных полностью онлайн, что позволяет пользователям платить только за используемые ресурсы. Это идеальный выбор для бизнеса или ведомственных приложений или витрин данных, а также отличная песочница для исследователей данных или разработчиков. Благодаря выделенной инфраструктуре клиенты получают свою собственную выделенную инфраструктуру Exadata в Oracle Cloud, что фактически дает им Частное облако баз данных в Oracle Public Cloud. Он работает внутри аппаратно принудительной виртуальной облачной сети, обеспечивая высочайший уровень изоляции от других арендаторов. Пользователи могут легко настроить одну или несколько контейнерных баз данных на своей выделенной инфраструктуре, каждая из которых может иметь одну или несколько подключаемых баз данных внутри нее. Выделенная инфраструктура позволяет настроить операционные политики, используемые для управления подготовкой новой базы данных, сроками обновления, доступностью и плотностью баз данных, которые они запускают в инфраструктуре. Наличие контроля над версиями баз данных и сроками обновления особенно важно для

					<i>27.03.04.2020.046.00.00 ПЗ ВКР</i>	<i>Лист</i>
<i>Изм.</i>	<i>Лист</i>	<i>№ докум.</i>	<i>Подпись</i>	<i>Дата</i>		<i>25</i>

приложений, которые более чувствительны к различиям версий и выпусков баз данных. Хотя клиенты отменяют эти операционные политики, все операции по-прежнему полностью автоматизированы Oracle. С помощью выделенной инфраструктуры клиенты имеют возможность определять администраторов "флота", которые управляют общей службой, а также отдельных лиц, которые могут развертывать базы данных и управлять ими. It is an идеальная платформа для клиентов которые хотят переосмыслить свою ИТ стратегию и перенести часть или всю свою базу данных в облачное решение.

Автономная база данных устраняет необходимость устанавливать и управлять сложными настройками конфигурации, которые управляют тем, как работают базы данных Oracle. Эти сложные настройки автоматически управляются Oracle, но ряд дополнительных настроек доступен клиентам, если они хотят настроить параметры локализации (NLS) и / или настройки оптимизатора SQL.

Форматы отображения даты, времени и валюты варьируются по всему миру и могут быть установлены клиентами в соответствии с потребностями их пользователей и приложений.

Хотя скриптовое масштабирование обеспечивает гибкость и контроль над масштабированием услуг, функция автоматического масштабирования обеспечивает возможность автоматического масштабирования в заранее определенных границах в ответ на требования рабочей нагрузки в режиме реального времени. Функция автоматического масштабирования feature begins with a baseline OCPU configuration and автоматически масштабирует процессоры до 3-кратного базового уровня при увеличении рабочей нагрузки. Базовая линия может быть изменена вручную или с помощью REST API, и автоматическое масштабирование будет работать от установленной базовой линии. Дополнительные процессоры используются только в случае необходимости, и автономная база данных мгновенно возвращается к базовому процессору, когда рабочая нагрузка больше не требует

					<i>27.03.04.2020.046.00.00 ПЗ ВКР</i>	<i>Лист</i>
<i>Изм.</i>	<i>Лист</i>	<i>№ докум.</i>	<i>Подпись</i>	<i>Дата</i>		<i>26</i>

дополнительных ресурсов. Остановка и запуск автономной службы автономные службы баз данных могут быть полностью остановлены, а выставление счетов за вычислительные ресурсы прекращается во время остановки службы. Остановка производства автономных баз данных не очень распространена, но эта возможность, безусловно, полезна для разработки, тестирования и других непродуцированных баз данных. QA (обеспечение качества) и обучающие базы данных также являются случаями, когда автономная база данных часто может быть отключена, когда служба не используется.

Постоянное ежедневное управление Автономной базой данных полностью автоматизировано. Управление пространством, исправление и обновление, настройка, диагностика ошибок и регулярная проверка работоспособности выполняются автоматически, не требуя вмешательства пользователя.

Автоматизированная база данных ManagementOracleAutonomous включает в себя полный набор продуктов, найденных в Oracle Exadata, и облачный сервис Exadata, но все компоненты работают автономно. Чтобы поставить это в перспективе, Оракул автономный базе лежат следующие компоненты:•Oracle в облачной инфраструктуре•Exadata серверов и устройств хранения данных•технологии InfiniBand сети•Exadata хранения серверного программного обеспечения•виртуальных машин Oracle•корпорация Oracle Grid-инфраструктуры•Oracle для реальных приложений кластеры•базы данных Oracle предприятия EditionAll этих компонентов осуществляется автоматически в Автономной базе данных, включая первоначальную настройку, изменения в ответ на масштабирование вверх/вниз требования, и во время службы удаление. В этом разделе описывается, как клиенты могут просматривать действия управления, которые были предприняты Oracle или запланированы к выполнению. В этом разделе также описывается управленческий контроль, который клиенты имеют над сервисом.Автоматическое обновление автоматическая база данных с

					<i>27.03.04.2020.046.00.00 ПЗ ВКР</i>	<i>Лист</i>
<i>Изм.</i>	<i>Лист</i>	<i>№ докум.</i>	<i>Подпись</i>	<i>Дата</i>		<i>27</i>

разделенной инфраструктурой автоматически обновляется до последних версий всех программных и базовых компонентов. Все обновления автоматически выполняются Oracle без сбоев в периоды низкого использования сервиса, и отдельным клиентам не нужно беспокоиться о сроках применения обновлений. Автономная база данных Rolling Upgrades & Application Continuity Oracle построена на базе кластеризованных систем с использованием Exadata, которая обеспечивает прокатные обновления. Рабочие нагрузки, соответствующие требованиям Oracle Application Continuity requirements and best practices, перемещаются из одного узла кластера в другой, не оказывая влияния на приложения или пользователей. Непрерывность приложения включает перемещение сеансов между узлами кластера даже во время выполнения транзакций. Транзакции часто могут продолжать обработку после перемещения на другой узел без потери состояния сеанса.

Для реализации клиентского приложения использован язык PHP 7.4.1.

Этот язык обладает рядом преимуществ:

- специально создан для web-разработок;
- сценарии выполняются на сервере и встраиваются непосредственно в HTML, что позволяет создавать динамично изменяемые HTML-страницы;
- язык прост, но многофункционален;
- поддерживает работу с большим количеством баз данных, в том числе ORACLE;

PHP – это кроссплатформенная технология, которая может использоваться на всех крупных операционных системах, включая Linux, многие варианты Unix (HP-UX, Solaris и OpenBSD), Microsoft Windows, Mac OS X, RISC OS и, другие;

PHP имеет поддержку для большинства существующих web-серверов. Это Apache, Microsoft Internet Information Server, Personal Web Server, Netscape и iPlanet-серверы и другие;

PHP позволяет использовать возможности как процедурного, так и объектно-ориентированного программирования;

					<i>27.03.04.2020.046.00.00 ПЗ ВКР</i>	<i>Лист</i>
<i>Изм.</i>	<i>Лист</i>	<i>№ докум.</i>	<i>Подпись</i>	<i>Дата</i>		<i>28</i>

PHP может автоматически генерировать файлы разнообразных форматов (изображения, PDF, XML и т.д.) и сохранять их в файловой системе;

PHP поддерживает взаимодействие с другими службами по таким протоколам, как NNTP, POP3, HTTP, COM (под Windows) и множество других.

Основные нововведения в версиях 7 и выше:

- Поддержка 64-разрядной Windows
- Удаление устаревших функций и расширений
- Анонимные классы (классы без имени, которые могут ускорить процесс кодирования, а также время выполнения)
- Унифицированный синтаксис переменных
- Синтаксис Unicode Codepoint Escape. Новый escape-символ (`\u`) дает возможность использовать код Unicode символа (в шестнадцатеричном виде) внутри строк.
- Общий родительский класс при обработке исключений (`\BaseException`)
- Декларация возвращаемого типа
- Абстрактное дерево синтаксиса. Это расширение позволяет получать промежуточное представление кода во время компиляции.
- Новый движок Zend Engine
- Производительность + Экономия памяти. PHP 7 работает в два раза быстрее, чем PHP 5.6 и использует значительно меньше памяти.

В первую очередь новшества в PHP 7 коснулись самых важных для разработки приложений параметров – это производительность и потребление памяти.

Основное изменение, о котором в первую очередь упоминают в разговоре о PHP 7, это **улучшение производительности**: показатели PHP 7 приближены к показателям HHVM (разработанной Facebook системы, которая позволяет ускорить преобразование PHP-кода в несколько раз). Не

					<i>27.03.04.2020.046.00.00 ПЗ ВКР</i>	<i>Лист</i>
<i>Изм.</i>	<i>Лист</i>	<i>№ докум.</i>	<i>Подпись</i>	<i>Дата</i>		<i>29</i>

последней причиной этого стало то, что соучредитель Zend Technologies (компания-разработчика ядра PHP) Зеев Сураски рассматривает HHVM как одного из конкурентов Zend. А ведущий разработчик PHP 7 компании Zend Technologies и автор phpng Дмитрий Стогов заверяет, что разница в производительности PHP 5.0 и PHP 7 на синтетическом тесте bench.php составляет 14 раз. На практике большинство пользователей склоняются к мнению, что PHP 7 примерно в два раза быстрее предыдущей версии.

HHVM использует так называемую JIT (Just-In-Time) компиляцию для преобразования кода – несмотря на то, что JIT в 7 версии PHP не появился, PHP 7 поддерживает более хорошую миграцию, которая и позволяет улучшить производительность в разы. Именно новый уровень производительности Дмитрий Стогов выделяет в качестве главной особенности PHP 7.

Улучшение производительности удалось добиться и благодаря тому, что компилятор теперь генерирует более эффективный и короткий байт-код.

В наше время крайне важно думать о пользователях не только стационарных, но и мобильных устройств, поэтому разработчики PHP 7 улучшили исполнения движка и снизили потребление памяти, а также добавили возможность использовать нативное локальное хранилище. К тому же PHP 7 позволяет выполнять асинхронные задачи.

Процесс установки PHP включает в себя следующие действия:

1) Необходимо скачать дистрибутив PHP на официальном сайте, выбрать нужную версию и скачать ZIP-архив для типа сборки Thread-Safe (чтобы он был совместим с сервером Apache) и подходящей разрядности. После того, как закончится загрузка, нужно распаковать архив в любую папку.

Проверка работы PHP

Для проверки будем использовать командную строку (консоль).

					27.03.04.2020.046.00.00 ПЗ ВКР	Лист
Изм.	Лист	№ докум.	Подпись	Дата		30

Из консоли нам нужно выбрать обработчик (в данном случае это файл `php.exe` в папке `C:\PHP7\`).

Проще всего сделать это в Windows 10 так: откройте File Explorer и перейдите в нужную папку (рисунок 4.2):

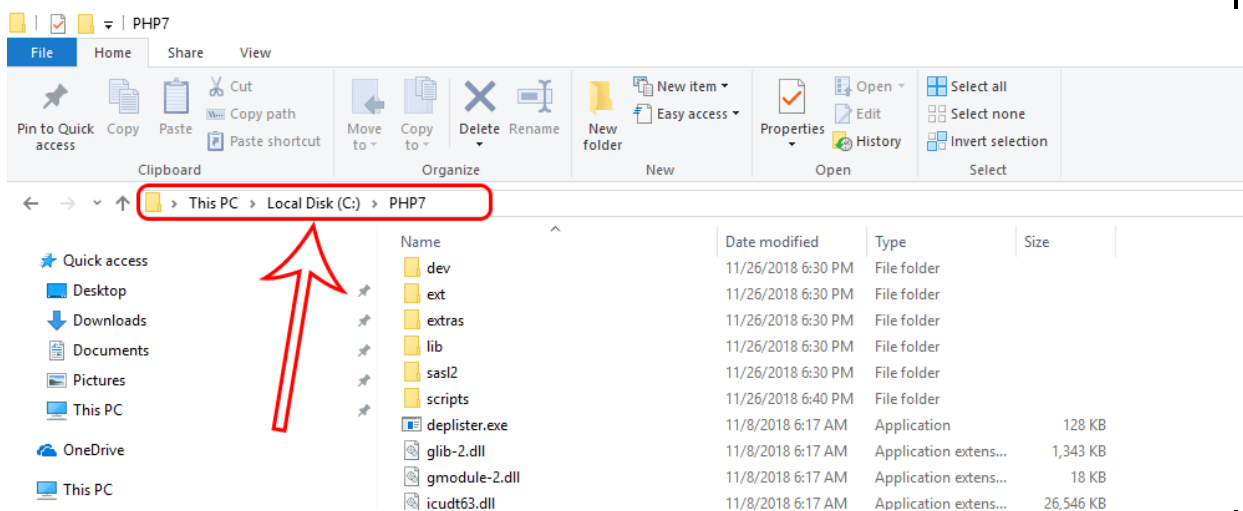


Рисунок 4.2 – Проверка работы PHP

Затем в строке пути к папке нужно набрать слово `cmd` и нажать `Enter`. Откроется командная строка (консоль; CLI).

Теперь нужно выбрать обработчик и скрипт для обработки, — для этого набрать следующую команду в консоли:

```
php.exe C:\PHP7\scripts\script_01.php
```

- `php.exe` – это обработчик. Поскольку мы запустили консоль из папки с этим файлом, путь к нему мы опустили. Если вы запускаете обработчик из другой папки, тогда указывайте полный путь к нему, например, в формате `C:\PHP7\php.exe`.
- Через пробел мы указываем скрипт для обработки (абсолютный путь к файлу скрипта) – `C:\PHP7\scripts\script_01.php`. (рисунок 4.3)

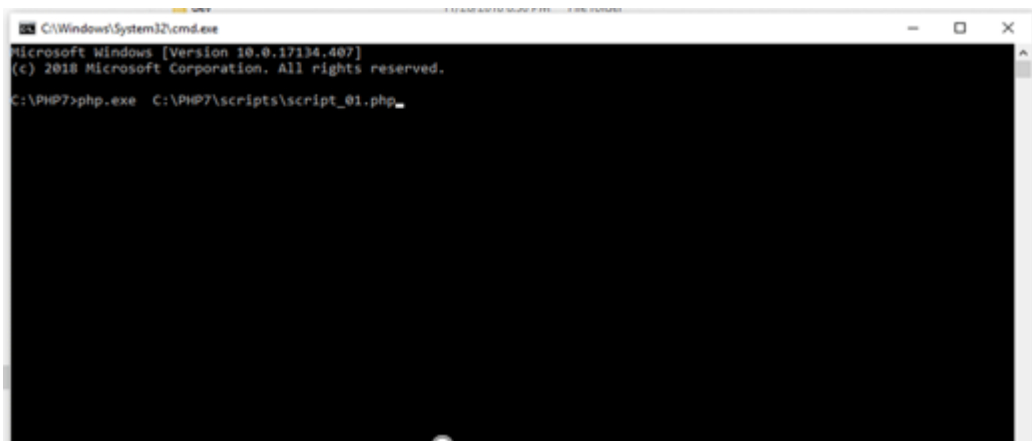


Рисунок 4.3

Использование встроенного в PHP сервера: если необходимо запускать PHP-скрипты из браузера, тогда нужно установить дополнительно веб-сервер (например, Apache). Для простой проверки можно обойтись и без Апаха, а использовать встроенный в дистрибутив PHP сервер.

Необходимо открыть в File Explorer папку с установленным PHP и набрать в пути слово `cmd`, а затем нажмите `Enter`.

После этого наберите команду, которая переведет PHP в режим сервера:

```
php.exe -S localhost:6789
```

- `-S` – эта часть команды обозначает, что PHP должен запускаться в режиме сервера
- `localhost` – эта часть команды обозначает, что сервер должен принимать соединения только со своего компьютера
- `6789` — это номер порта, на котором сервер будет ждать соединения от веб-браузера. При ошибке о занятости порта, используйте другой порт (от 1 до 65534).

После того как вы наберете команду и нажмете `Enter`, вы должны увидеть следующую информацию в консоли (рисунок 4.4):

- Версию запущенного PHP и время запуска
- Порт, который слушает сервер

					<i>27.03.04.2020.046.00.00 ПЗ ВКР</i>	Лист
Изм.	Лист	№ докум.	Подпись	Дата		32

- Корневая папка сервера (в нашем примере это C:\PHP7\)
- Чтобы завершить работу сервера, вы должны нажать в консоли Ctrl+C

```

C:\Windows\System32\cmd.exe - php.exe -S localhost:6789
Microsoft Windows [Version 10.0.17134.407]
(c) 2018 Microsoft Corporation. All rights reserved.

C:\PHP7>php.exe -S localhost:6789
PHP 7.2.12 Development Server started at Mon Nov 26 19:52:07 2018
Listening on http://localhost:6789
Document root is C:\PHP7
Press Ctrl-C to quit.

```

Рисунок 4.4 – Информация в консоли

В PHP 7 есть множество новых функций, каждая из которых делает разработку более удобной и простой.

- Два новых класса исключений (Exception и Error) – если в предыдущих версиях PHP нельзя было обрабатывать фатальные ошибки, то в новом релизе приводящие к этому действия выбрасывают исключения, то есть завершение скрипта не произойдет.

- Анонимные классы – теперь PHP 7 поддерживает этот вид классов, который часто используют в C++ и Java для выполнения callback-функций. Эти классы в основном используются для создания простых элементов.

Анонимные классы могут быть вложенными, но при этом им не будут доступны protected или private-свойства внешнего класса.

- Возможность указывать скалярные типы (Scalar Type Hints) - в предыдущих версиях PHP можно было указывать тип принимаемых значений; теперь вы также можете указывать и скалярные типы:

строки (``string``),
целые (``int``),

					<i>27.03.04.2020.046.00.00 ПЗ ВКР</i>	Лист
Изм.	Лист	№ докум.	Подпись	Дата		33

числа (``float``),

бульки (``bool``).

Это нововведение направлено на облегчение написания кода и улучшение его читабельности. А вот использование этих значений в качестве имен классов категорически запрещено, т.к. это может привести к критическим ошибкам.

- Сокращенная конструкция `use` - в PHP 7 появилась возможность объединять декларирование групп `use`, что, несомненно, положительно отразится как на скорости работы, так и на восприятии кода (таблица 1):

Таблица 1 – Улучшения в PHP 7.

Было в предыдущих версиях	Стало в PHP 7
<pre>use function some\namespace\fn_a; use function some\namespace\fn_b; use function some\namespace\fn_c;</pre>	<pre>use function some\namespace\{ fn_a, fn_b, fn_c };</pre>

- Новые операторы Null coalescing operator (`??`) и Combined Comparison Operator (`<=>`, также он известен как Spaceship Operator) – эти два новых оператора, которых некоторые относят к синтаксическому сахару, помогут написать более лаконичный и понятный код.

Null coalescing operator – это оператор объединения, который проверяет, существует ли переменная, и возвращает ее значение (либо значение по умолчанию):

```
$action = $_POST['action'] ?? 'index';
```

Combined Comparison Operator – это оператор сравнения, который пригодится для использования в callback-функциях для `usort()`. Он сравнивает два значения и, в зависимости от результата, возвращает -1, 0 или 1:

```
function order_func($a, $b) {
    return $a <=> $b; }
```

Приоритет данной операции равен приоритетам других операций сравнения, при этом его нельзя связать с предыдущим вызовом, т.к. это неассоциативный оператор.

- Функция `assert()` - в PHP 7 эта функция обрабатывается особым образом для того, чтобы были исключены накладные расходы; вы можете использовать ее для отладки, а в дальнейшем отключать.

- Фильтрация `unserialize()` – это новшество обеспечит безопасность сериализации объектов. Разработчик сам сможет определить, какие классы должны подвергнуться обработке, а какие нет.

- Представление целых чисел в виде 64-битных в PHP 7 для Windows x64: теперь PHP поддерживает строки длиной больше 231 байт в 64-битных сборках.

- Делегирование генераторов – при помощи `<expr>` сложный генератор можно разделить на несколько более мелких. Как и некоторые другие нововведения, данное новшество направлено на написание более чистого и удобного для повторного использования кода.

- Появление нового класса `IntlChar` – он увеличивает набор возможностей библиотеки для локализации приложений International Components for Unicode и имеет методы класса и константы для работы с Unicode.

- Возможность определять массивы как значения констант, которые объявляются через `define()`.

- Возможность присваивать зарезервированные ключевые слова именам методов.

- Появление нового идентификатора простых значений `PHP_INT_MIN`.

Что касается старых функций, то часть самых не востребуемых возможностей была исключена: к примеру, были удалены ASP и script-теги, а также регулярные выражения, совместимые с POSIX, и расширение `ext/mysql`.

					<i>27.03.04.2020.046.00.00 ПЗ ВКР</i>	<i>Лист</i>
<i>Изм.</i>	<i>Лист</i>	<i>№ докум.</i>	<i>Подпись</i>	<i>Дата</i>		<i>35</i>

В качестве web-сервера приложения использован Apache 2.2.10.

Apache – это веб-сервер под Windows. На веб-сервере можно размещать сайты и открывать их на своём или на других компьютерах так, будто они находятся в Интернете.

Apache является бесплатным, у него открыт исходный код, сервер гибок в настройке.

Настройка Apache выполняется в текстовых файлах, а для запуска нужно использовать командную строку – для пользователей Windows это может оказаться непривычным. Ситуация осложняется тем, что подсказки в конфигурационных файлах на английском языке, а веб-сервер в случае ошибки выводит очень мало информации.

Настройка Apache выполняется следующим образом.

Установим минимально необходимые для работы атрибуты, для этого откроем для редактирования файл `/etc/httpd/conf/httpd.conf`

Apache может работать с одним из трех доступных мультипроцессорных модулей (mpm):

1. *prefork* — один процесс на запрос, наиболее часто используемый модуль;
2. *worker* — несколько процессов с несколькими нитями у каждого;
3. *event* — новый экспериментальный вариант стандартного *worker* мультипроцессорного модуля.

Для просмотра информации о модулях выполните:

```
# httpd -V | grep "Server MPM"
```

ПРИМЕР КОНФИГУРАЦИИ ДЛЯ АРАСНЕ 2.4

```
<IfModule mpm_prefork_module>
```

```
ServerLimit 256
```

```
StartServers 20
```

```
MinSpareServers 20
```

```
MaxSpareServers 20
```

```
MaxRequestWorkers 256
```

					27.03.04.2020.046.00.00 ПЗ ВКР	Лист
Изм.	Лист	№ докум.	Подпись	Дата		36

MaxConnectionsPerChild 4000

</IfModule>

MaxRequestWorkers установите значением, равным *ServerLimit*.

Пример конфигурации для Apache 2

В секции <IfModule [мультипроцессорный модуль]>, например, <IfModule prefork.c> установим следующие значения:

```
StartServers    5
MinSpareServers 10
MaxSpareServers 10
MaxClients     30
ServerLimit    30
```

StartServers — количество запускаемых процессов, рассчитывается как (Всего памяти - Память для сервера, MySQL, Memcache и т.д.) / размер процесса. Например, 64Gb - 32Gb / 0.256Gb = максимум 125 процессов.

MinSpareServers — минимальное количество простаивающих процессов.

MaxSpareServers — максимальное количество простаивающих процессов.

MaxClients — лимит процессов.

ServerLimit — устанавливается аналогичным значением *MaxClients*.

Обстоятельно необходимо подходить к расчету *MaxClients*, т.к. при установке малого значения может увеличиться количество отказов веб-сервера в обслуживании на запрос клиента. Установка слишком большого значения может привести к свопингу и значительному увеличению времени ответа веб-сервера.

При использовании одноуровневой и двухуровневой конфигурации веб-сервера лимиты устанавливаются по-разному. Отдавая сформированные страницы более легкому веб-серверу (например, nginx), лимиты можно понизить. Конкретные значения рассчитываются индивидуально для каждого веб-сервера.

					27.03.04.2020.046.00.00 ПЗ ВКР	Лист
Изм.	Лист	№ докум.	Подпись	Дата		37

Интеграция с другим ПО и языками программирования

Существует множество модулей, добавляющих к Apache поддержку различных языков программирования и систем разработки.

К ним относятся:

- PHP (mod_php).
- Python (mod_python, mod_wsgi).
- Ruby (apache-ruby).
- Perl (mod_perl).
- ASP (apache-asp)^[13].
- Tcl (rivet)^[14]

Кроме того, Apache поддерживает механизмы CGI и FastCGI, что позволяет исполнять программы на практически всех языках программирования, в том числе C, C++, Lua, sh, Java.

Apache имеет различные механизмы обеспечения безопасности и разграничения доступа к данным. Основными являются:

- Ограничение доступа к определённым каталогам или файлам.
- Механизм авторизации пользователей для доступа к каталогу на основе HTTP-аутентификации (mod_auth_basic) и digest-аутентификации (mod_auth_digest).
- Ограничение доступа к определённым каталогам или всему серверу, основанное на IP-адресах пользователей.
- Запрет доступа к определённым типам файлов для всех или части пользователей, например запрет доступа к конфигурационным файлам и файлам баз данных.
- Существуют модули, реализующие авторизацию через СУБД или РАМ.

В некоторых MPM-модулях присутствует возможность запуска каждого процесса Apache, используя различные uid и gid с соответствующими этим пользователям и группам пользователей.

					27.03.04.2020.046.00.00 ПЗ ВКР	Лист
Изм.	Лист	№ докум.	Подпись	Дата		38

Также существует механизм suexec, используемый для запуска скриптов и CGI-приложений с правами и идентификационными данными пользователя.

Для реализации шифрования данных, передающихся между клиентом и сервером, используется механизм SSL, реализованный через библиотеку OpenSSL. Для удостоверения подлинности веб-сервера используются сертификаты X.509.

Существуют внешние средства обеспечения безопасности, например mod_security.

Для создания и использования системы Service Desk не потребуется установки дополнительных автоматизированных рабочих мест. Достаточно ПЭВМ, подключенной к компьютерной сети предприятия.

					<i>27.03.04.2020.046.00.00 ПЗ ВКР</i>	<i>Лист</i>
<i>Изм.</i>	<i>Лист</i>	<i>№ докум.</i>	<i>Подпись</i>	<i>Дата</i>		<i>39</i>

5 РАЗРАБОТКА БЛОК-СХЕМ АЛГОРИТМОВ

Алгоритм — точный набор инструкций, описывающих порядок действий исполнителя для достижения результата решения задачи за конечное время. Блок-схемой называют графическое представление алгоритма, в котором он изображается в виде последовательности связанных между собой функциональных блоков, каждый из которых соответствует выполнению одного или нескольких действий. [1]

В ходе работы программы операции выполняются по определенному алгоритму. Рассмотрим блок-схему алгоритма регистрации инцидента (рисунок 5.1).

					<i>27.03.04.2020.046.00.00 ПЗ ВКР</i>	<i>Лист</i>
<i>Изм.</i>	<i>Лист</i>	<i>№ докум.</i>	<i>Подпись</i>	<i>Дата</i>		<i>40</i>

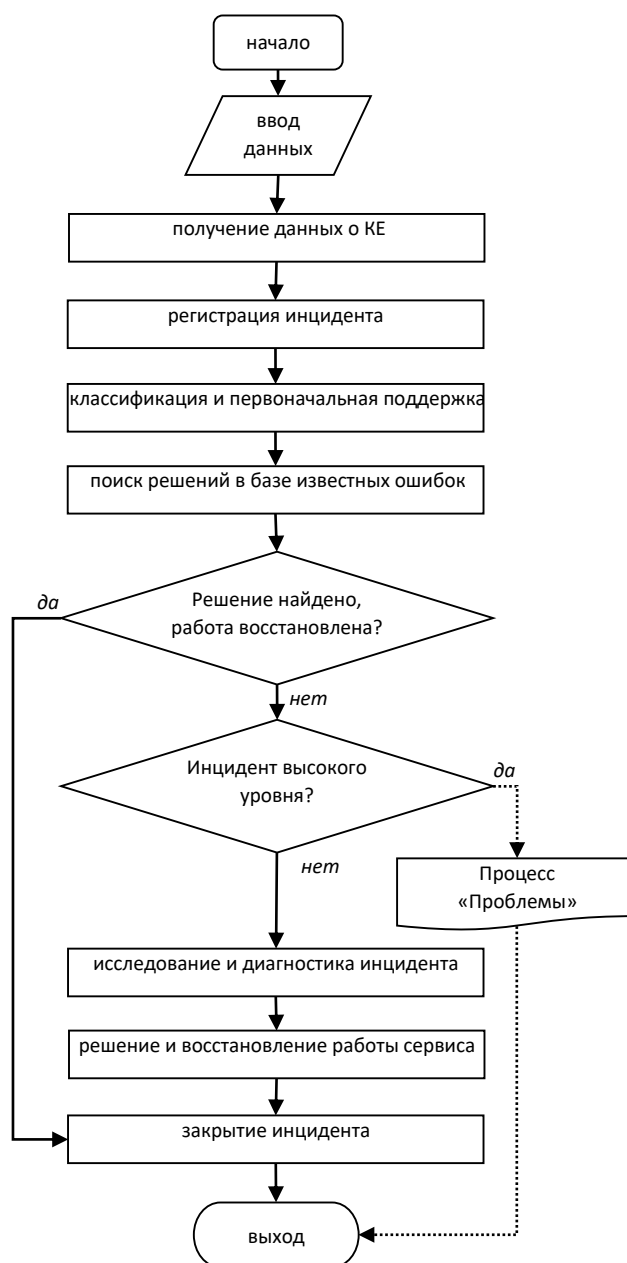


Рисунок 5.1 – Блок-схема алгоритма регистрации инцидента

Процесс «Инциденты» содержит следующие этапы:

1. Диспетчер получает данные от пользователя и вводит их в журнал регистрации инцидентов.
2. Диспетчер получает данных о КЕ.
3. Диспетчер регистрирует инцидент в системе на основании данных, полученных от пользователя:

наименование подразделения;

контактное лицо (ФИО, должность, номер телефона, адрес электронной почты);

сведения о КЕ;

описание сбоя;

предполагаемый приоритет запроса (в порядке очереди, важно, очень важно).

4. Согласно полученным данным и на основании сведений, полученных из БД «КЕ» диспетчер классифицирует инцидент.

5. Диспетчер осуществляет поиск решения в базе известных ошибок.

6. Если решение найдено, диспетчер, убедившись в восстановлении сервиса, закрывает инцидент (переход к п. 11).

7. Если известное решение отсутствует, диспетчер направляет инцидент специалисту по направлению.

8. Если инцидент высокого уровня серьезностью, то специалист по направлению инициирует данный инцидент в проблему.

9. Если инцидент невысокого уровня серьезностью, то специалист по направлению исследует и диагностирует инцидент.

10. Специалист по направлению находит решение и восстанавливает работу сервиса.

11. Специалист по направлению, убедившись в восстановлении сервиса, закрывает инцидент.

Процесс «Проблемы» содержит следующие этапы:

1. Владелец проблемы осуществляет контекстный поиск существующих проблем.

2. Если в БД существующая проблема уже зафиксирована, специалист обновляет запись об инциденте ссылкой на проблему.

3. Если проблема не найдена в БД, специалист создает запись о новой проблеме и обновляет запись об инциденте ссылкой на проблему.

					27.03.04.2020.046.00.00 ПЗ ВКР	Лист
Изм.	Лист	№ докум.	Подпись	Дата		42

4. Специалист по направлению исследует проблему, находит ее причину.

5. Если причина найдена, управление передается процессу «Известные ошибки».

6. Специалист по направлению закрывает проблему и связанные инциденты.

Процесс «Ошибки» содержит следующие этапы:

1. Специалист создает запись о новой известной ошибке.
2. Специалист по направлению исследует ошибку, находит возможные пути исправления.
3. Если пути исправления найдены, специалист закрывает.

Процесс «Изменения» содержит следующие этапы:

1. Ввод данных о пользователе (источник – заявка на изменение).
2. Получение данных о КЕ (источник БД КЕ).
3. Ввод данных об изменении (источник – заявка на изменение).

Процесс «Конфигурации» содержит следующие этапы:

1. Получение данных о возможных составляющих конфигураций (источник – БД конфигураций).
2. Регистрация новой конфигурации при необходимости.

Процесс «Регистрация КЕ» содержит следующие этапы:

1. Получение данных о пользователе (источник – БД пользователей).
2. получение данных о конфигурации (источник – БД КЕ).
3. Ввод данных о КЕ (источник – заявка на подключение).

Процесс «Отчеты» содержит следующие этапы:

					<i>27.03.04.2020.046.00.00 ПЗ ВКР</i>	<i>Лист</i>
<i>Изм.</i>	<i>Лист</i>	<i>№ докум.</i>	<i>Подпись</i>	<i>Дата</i>		<i>43</i>

1. Ввод или выбор данных, необходимых для формирования условий отчета.

2. Получение данных отчета (источник – БД КЕ, БД пользователей, БД инцидентов, БД проблем, БД ошибок, БД изменений, БД сотрудников).

Общая блок-схема алгоритма поддержки пользователей показана на рисунке 5.2.

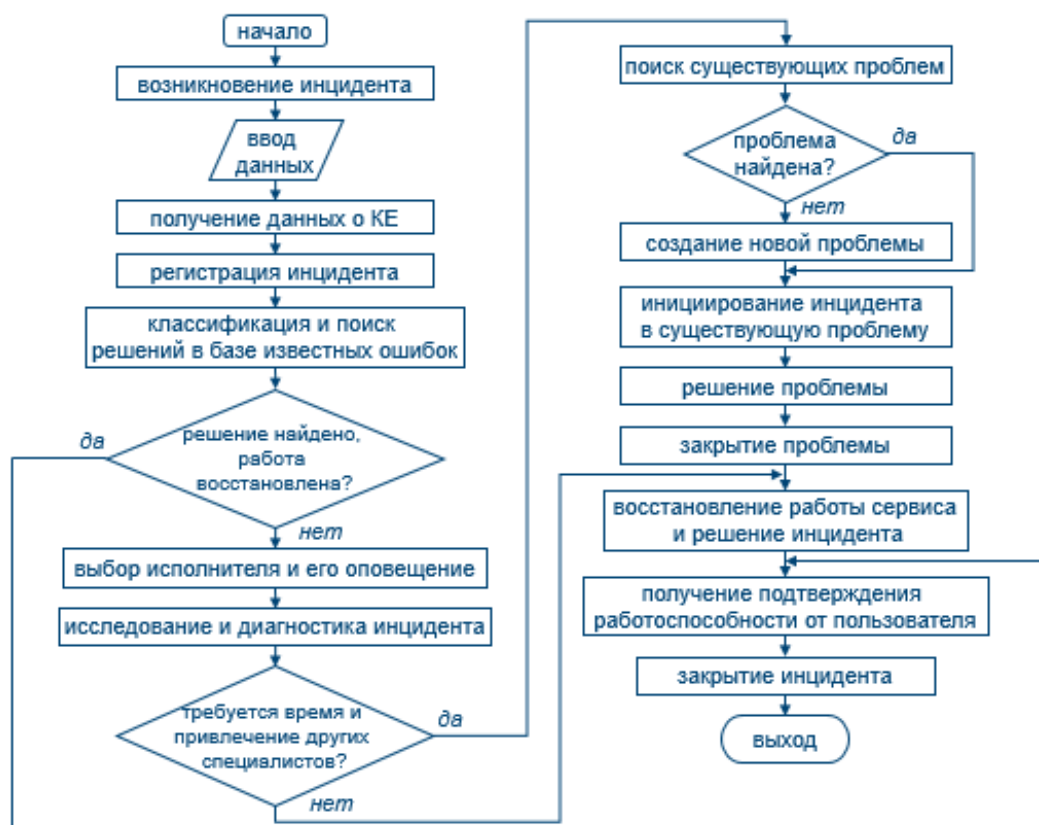


Рисунок 5.2 – Блок-схема алгоритма поддержки пользователей

6 ИНФОРМАЦИОННОЕ И ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ КОМПЛЕКСА ЗАДАЧ

Определим основные понятия предметной области:

ИТ-услуга (ИТ-сервис) – услуга, предоставляемая ИТ-подразделением пользователям.

Конфигурация – стандарт ИТ-сервиса.

Справочник конфигураций – перечень принятых к использованию конфигураций.

Конфигурационная единица (КЕ) – элемент ИТ-инфраструктуры (единица оборудования, учетная запись пользователя, документация и т.д.).

Пользователь – потребитель ИТ-услуги.

Рабочее место – конфигурационная единица в привязке к пользователю.

Инцидент – сбой в работе ИТ-сервиса.

Проблема – один или несколько инцидентов, причина которых неизвестна.

Известная ошибка – проблема, причина которой установлена.

Изменение – любое изменение свойств КЕ.

Идентифицируем связи между понятиями:

Конфигурации используются при предоставлении ИТ-сервисов.

Пользователь использует КЕ.

Конфигурация определяет стандарт, используемый КЕ.

КЕ является объектом инцидента.

Владелец в УИТ сопровождает инцидент.

КЕ является объектом проблемы.

Владелец в УИТ сопровождает проблему.

Владелец в УИТ сопровождает известную ошибку.

КЕ является объектом изменения.

Владелец в УИТ сопровождает изменение.

Проблема инициирует известную ошибку.

					<i>27.03.04.2020.046.00.00 ПЗ ВКР</i>	<i>Лист</i>
<i>Изм.</i>	<i>Лист</i>	<i>№ докум.</i>	<i>Подпись</i>	<i>Дата</i>		<i>45</i>

Известная ошибка инициирует Изменение.

Модель данных, построенная на основании выделенных сущностей и связей, представлена на рисунке 6.1.

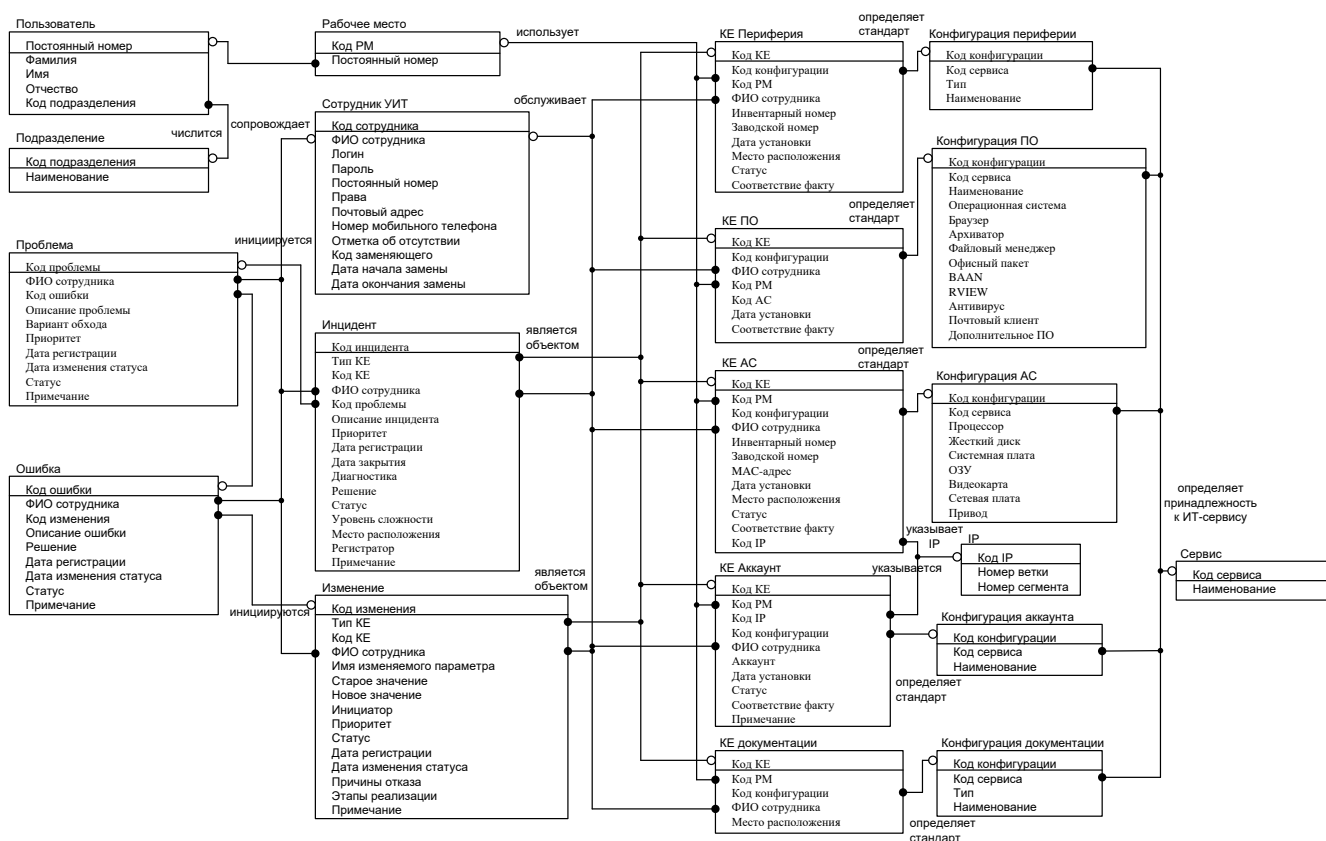


Рисунок 6.1 – Логическая модель данных

Физический уровень реализуется в виде модели СУБД. В ходе определения основных объектов предметной области, был выделен объект «Конфигурационная единица». Так как этот объект объединяет разнородные элементы, имеющие также ряд схожих атрибутов, в логической модели данных была сформирована иерархия наследования с родовым предком «KE».

Создадим модель СУБД. Известно, что в нормализованной БД для выполнения транзакций или запросов более интенсивно используется центральный процессор, требуется больше памяти и большее число операций ввода-вывода, чем в ненормализованной, так как для извлечения и обработки данных приходится находить и связывать соответствующие таблицы и записи.

С учетом того, что на физическом уровне излишняя нормализация данных может привести к замедлению работы проведем процесс денормализации.

С целью уменьшения числа связываемых таблиц при доступе к данным, исключим при создании физической БД сущность «КЕ». Атрибуты этой сущности разнесем следующим образом: тип КЕ будем хранить в таблицах «INCIDENTS» И «CHANGES», а ФИО сотрудника, код рабочего места и статус в таблицах каждого типа КЕ «PERIPHERY», «SOFT», «HARD», «ACCOUNT», «DOC» (рисунок 6.2).

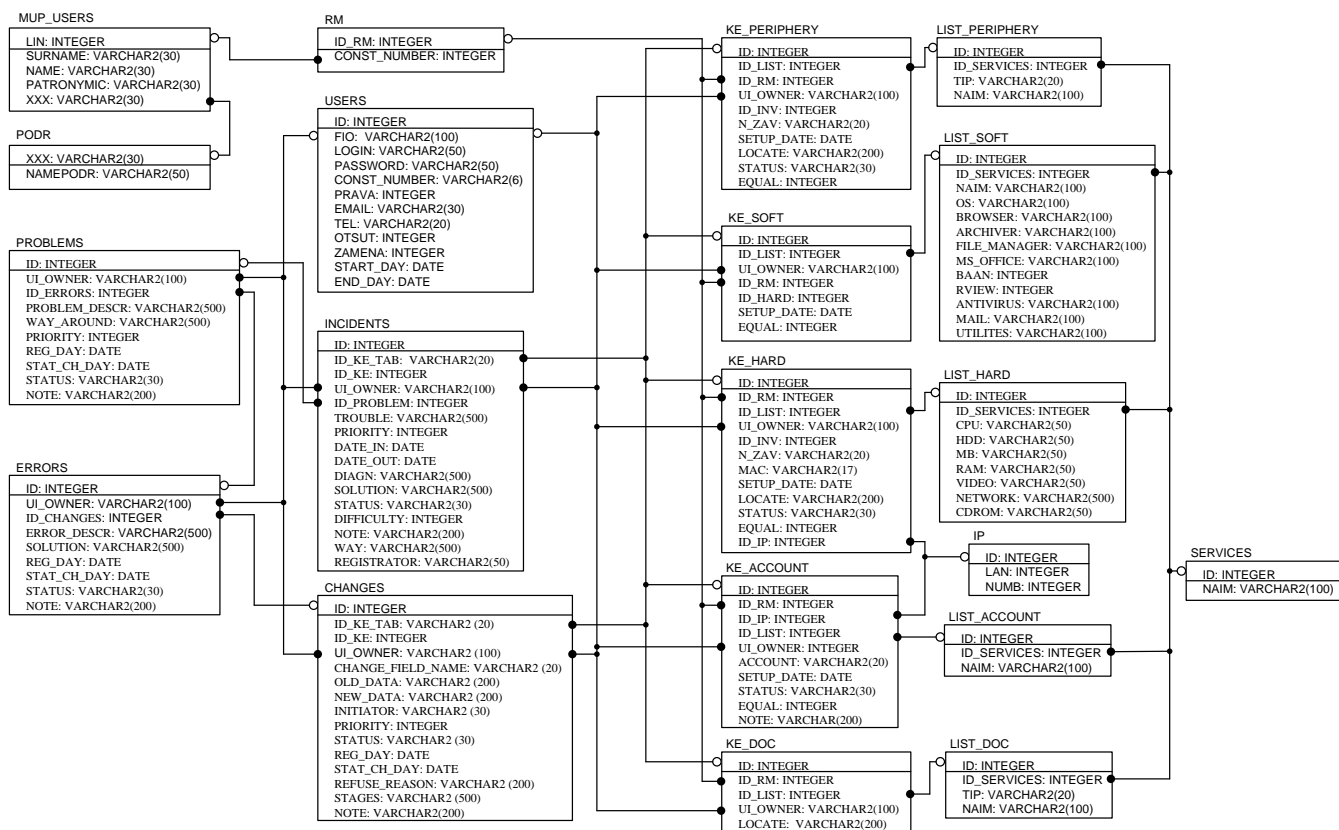


Рисунок 6.2 - Физическая модель СУБД

27.03.04.2020.046.00.00 ПЗ ВКР

Лист

47

Изм.	Лист	№ докум.	Подпись	Дата
------	------	----------	---------	------

Таблица 2 – «HARD»

Имя атрибута	Имя поля	Тип данных
Код КЕ	ID	INTEGER
Код конфигурации	ID_LIST	INTEGER
Код РМ	ID_RM	INTEGER
ФИО сотрудника	UI_OWNER	VARCHAR2(100)
Инвентарный номер	ID_INV	INTEGER
Заводской номер	N_ZAV	VARCHAR2(20)
MAC-адрес	MAC	VARCHAR2(17)
Дата установки	SETUP_DATE	DATE
Место расположения	LOCATE	VARCHAR2(200)
Статус	STATUS	VARCHAR2(30)
Соответствие факту	EQUAL	INTEGER
Код IP	ID_IP	INTEGER

Таблица 3 – «SOFT»

Имя атрибута	Имя поля	Тип данных
Код КЕ	ID	INTEGER
Код конфигурации	ID_LIST	INTEGER
ФИО сотрудника	UI_OWNER	VARCHAR2(100)
Код РМ	ID_RM	INTEGER
Код АС	ID_HARD	INTEGER
Дата установки	SETUP_DATE	DATE
Соответствие факту	EQUAL	INTEGER

Таблица 4 – «ACCOUNT»

Имя атрибута	Имя поля	Тип данных
Код КЕ	ID	INTEGER
Код РМ	ID_RM	INTEGER
Код конфигурации	ID_LIST	INTEGER
ФИО сотрудника	UI_OWNER	VARCHAR2(100)
Аккаунт	ACCOUNT	VARCHAR2(20)

Изм.	Лист	№ докум.	Подпись	Дата

27.03.04.2020.046.00.00 ПЗ ВКР

Продолжение таблицы 4 – «ACCOUNT»

Имя атрибута	Имя поля	Тип данных
Дата установки	SETUP_DATE	DATE
Статус	STATUS	VARCHAR2(30)
Соответствие факту	EQUAL	INTEGER
Код IP	ID_IP	INTEGER
Примечание	NOTE	VARCHAR2(200)

Таблица 5 – «PERIPHERY»

Имя атрибута	Имя поля	Тип данных
Код KE	ID	INTEGER
Код конфигурации	ID_LIST	INTEGER
Код РМ	ID_RM	INTEGER
ФИО сотрудника	UI_OWNER	VARCHAR2(100)
Инвентарный номер	ID_INV	INTEGER
Заводской номер	N_ZAV	VARCHAR2(20)
Дата установки	SETUP_DATE	DATE
Место расположения	LOCATE	VARCHAR2(200)
Статус	STATUS	VARCHAR2(30)
Соответствие факту	EQUAL	INTEGER

Таблица 6 – «DOC»

Имя атрибута	Имя поля	Тип данных
Код KE	ID	INTEGER
Код конфигурации	ID_LIST	INTEGER
Код РМ	ID_RM	INTEGER
ФИО сотрудника	UI_OWNER	VARCHAR2(100)
Место расположения	LOCATE	VARCHAR2(200)

Изм.	Лист	№ докум.	Подпись	Дата

27.03.04.2020.046.00.00 ПЗ ВКР

Лист

49

Выше определена следующую связь: KE содержит информацию о пользователе и конфигурации. Это касается всех перечисленных таблиц. Для того чтобы избежать избыточности, выносим данные о конфигурациях и пользователях в отдельные таблицы. Справочников конфигураций (таблицы 6 – 10) будет столько же, сколько типов KE.

Таблица 7 – «LIST_HARD»

Имя атрибута	Имя поля	Тип данных	Пример
Код конфигурации	ID	INTEGER	1
Код сервиса	ID_SERVICES	INTEGER	2
Процессор	CPU	VARCHAR2(50)	'Core i3-6100'
Жесткий диск	HDD	VARCHAR2(50)	'500GB'
Системная плата	MB	VARCHAR2(50)	'GA-B150M-HD3'
ОЗУ	RAM	VARCHAR2(50)	'4gb DDR3'
Видеокарта	VIDEO	VARCHAR2(50)	'Nvidia GE Force Fx5500'
Сетевая плата	NETWORK	VARCHAR2(500)	'I350-M4 GBA'
Привод	CDROM	VARCHAR2(50)	'отсутствует'

Таблица 8 – «LIST_SOFT»

Имя атрибута	Имя поля	Тип данных	Пример
Код конфигурации	ID	INTEGER	1
Код сервиса	ID_SERVICES	INTEGER	2
Наименование	NAIM	VARCHAR2(100)	'Стандарт1'
Операционная система	OS	VARCHAR2(100)	'Windows 7'
Браузер	BROWSER	VARCHAR2(100)	'Internet Explorer 9 sp1'
Архиватор	ARCHIVER	VARCHAR2(100)	'WinRAR3.0'
Файловый менеджер	FILE_MANAGE	VARCHAR2(100)	'Total Commander'

Изм.	Лист	№ докум.	Подпись	Дата

27.03.04.2020.046.00.00 ПЗ ВКР

Лист
50

Офисный пакет	MS_OFFICE	VARCHAR2(100)	'MS Office 2003'
RVIEW	RVIEW	INTEGER	0
Антивирус	ANTIVIRUS	VARCHAR2(100)	'Symantec'
Почтовый клиент	MAIL	VARCHAR2(100)	'Microsoft Outlook'
Дополнительные ПО	UTILITES	VARCHAR2(100)	''

Таблица 9 – «LIST_ACCOUNT»

Имя атрибута	Имя поля	Тип данных	Пример
Код конфигурации	ID	INTEGER	1
Код сервиса	ID_SERVICES	INTEGER	2
Наименование	NAIM	VARCHAR2(100)	'Подключение к Интернет'

Таблица 10 – «LIST_PERIPHERY»

Имя атрибута	Имя поля	Тип данных	Пример
Код конфигурации	ID	INTEGER	1
Код сервиса	ID_SERVICES	INTEGER	2
Тип	TIP	VARCHAR2(20)	'Принтер'
Наименование	NAIM	VARCHAR2(100)	'HP LaserJet 1100'

Таблица 11 – «LIST_DOC»

Имя атрибута	Имя поля	Тип данных	Пример
Код конфигурации	ID	INTEGER	1
Код сервиса	ID_SERVICES	INTEGER	2
Тип	TIP	VARCHAR2(20)	'Инструкция пользователя'
Наименование	NAIM	VARCHAR2(100)	'Инструкция пользователя SD'

Каждая конфигурация используется при предоставлении того или иного ИТ-сервиса. Данные о сервисах содержатся в таблице «SERVICES».

Таблица 12 – «SERVICES»

Имя атрибута	Имя поля	Тип данных	Пример
Код сервиса	ID	INTEGER	1
Наименование	NAIM	VARCHAR2(100)	‘Обслуживание ПО’

Таблицы «HARD» и «ACCOUNT» содержат одинаковый атрибут «ID_IP», который является ссылочным на IP адрес из таблицы «IP».

Таблица 13 – «IP»

Имя атрибута	Имя поля	Тип данных	Пример
Код IP	ID	Integer	1
Номер ветки	LAN	Integer	2
Номер сегмента	NUMB	Integer	3

Данные о пользователях хранятся в таблицах в виде кода рабочего места. Хотя каждый пользователь имеет уникальный постоянный номер, по причине возможности совмещения должностей, в том числе в различных подразделениях, этот параметр не подходит для использования в качестве уникального ключа. Исходя из этого, используется отдельная таблица «KRM», в которой устанавливаются уникальные коды для рабочих мест.

Таблица 14 – «KRM»

Имя атрибута	Имя поля	Тип данных	Пример
Код РМ	ID_RM	INTEGER	123
Постоянный номер	CONST_NUMBER	INTEGER	123456

Для описания объектов «Инцидент», «Проблема», «Известная ошибка», «Изменение» используются следующие таблицы:

Таблица 15 – «INCIDENTS»

Имя атрибута	Имя поля	Тип данных
Код инцидента	ID	INTEGER
Тип КЕ	ID_KE_TAB	VARCHAR2(20)
Код КЕ	ID_KE	INTEGER
ФИО сотрудника	UI_OWNER	VARCHAR2(100)
Код проблемы	ID_PROBLEM	INTEGER
Описание инцидента	TROUBLE	VARCHAR2(500)
Приоритет	PRIORITY	INTEGER
Дата регистрации	DATE_IN	DATE
Дата закрытия	DATE_OUT	DATE
Диагностика	DIAGN	VARCHAR2(500)
Решение	SOLUTION	VARCHAR2(500)
Статус	STATUS	VARCHAR2(30)
Уровень сложности	DIFFICULTY	INTEGER
Территория, где случился инцидент	WAY	VARCHAR2(500)
Регистратор	REGISTRATOR	VARCHAR2(50)
Примечание	NOTE	VARCHAR2(200)

Изм.	Лист	№ докум.	Подпись	Дата

27.03.04.2020.046.00.00 ПЗ ВКР

Лист

53

Таблица 16 – «PROBLEMS»

Имя атрибута	Имя поля	Тип данных
Код проблемы	ID	INTEGER
ФИО сотрудника	UI_OWNER	VARCHAR2(100)
Код ошибки	ID_ERRORS	INTEGER
Описание проблемы	PROBLEM_DESCR	VARCHAR2(500)
Вариант обхода	WAY_AROUND	VARCHAR2(500)
Приоритет	PRIORITY	INTEGER
Дата регистрации	REG_DAY	DATE
Дата изменения статуса	STAT_CH_DAY	DATE
Статус	STATUS	VARCHAR2(30)
Примечание	NOTE	VARCHAR2(200)

Таблица 17 – «ERRORS»

Имя атрибута	Имя поля	Тип данных
1	2	3
Код ошибки	ID	INTEGER
ФИО сотрудника	UI_OWNER	VARCHAR2(100)
Код изменения	ID_CHANGES	INTEGER
Описание ошибки	ERROR_DESCR	VARCHAR2(500)
Решение	SOLUTION	VARCHAR2(500)
Дата регистрации	REG_DAY	DATE
Дата изменения статуса	STAT_CH_DAY	DATE
Статус	STATUS	VARCHAR2(30)
Примечание	NOTE	VARCHAR2(200)

Изм.	Лист	№ докум.	Подпись	Дата

27.03.04.2020.046.00.00 ПЗ ВКР

Лист

54

Таблица 18 – «CHANGES»

Имя атрибута	Имя поля	Тип данных
1	2	3
Код изменения	ID	INTEGER
Тип KE	ID_KE_TAB	VARCHAR2 (20)
Код KE	ID_KE	INTEGER
Код сотрудника	UI_OWNER	VARCHAR2 (100)
Имя изменяемого параметра	CHANGE_FIELD_NAME	VARCHAR2 (20)
Старое значение	OLD_DATA	VARCHAR2 (200)
Новое значение	NEW_DATA	VARCHAR2 (200)
Инициатор	INITIATOR	VARCHAR2 (30)
Приоритет	PRIORITY	INTEGER
Статус	STATUS	VARCHAR2 (30)
Дата регистрации	REG_DAY	DATE
Дата изменения статуса	STAT_CH_DAY	DATE
Причины отказа	REFUSE_REASON	VARCHAR2 (200)
Этапы реализации	STAGES	VARCHAR2 (500)
Примечание	NOTE	VARCHAR2 (200)

Программно система реализована с применением технологии «клиент-сервер». Для работы в системе пользователю необходимо набрать адрес системы в адресной строке браузера, ввести логин и пароль на первой странице (рисунок 6.3).



Рисунок 6.3 – Окно ввода имя пользователя и пароля

После получения доступа, пользователь попадает на главную страницу, где размещаются инциденты, которые находятся у него в работе.

В верхней части окна программы располагается меню, подпункты которого имеют вид «всплывающих» строк.

На рисунке 6.4 показана структура меню системы Service Desk.



Рисунок 6.4 – Структура меню

Регистрация инцидента происходит в разделе Инциденты->Регистрация.

На первом этапе необходимо определить КЕ, с которой случился сбой. Для этого необходимо выбрать в списке тип КЕ и задать условия для поиска. В качестве критерия могут выступать ФИО пользователя, его постоянный номер, а также для учетных записей и АС имя учетной записи или инвентарный номер оборудования соответственно. После того как КЕ найдена, вносятся данные о сбое, если инцидент решен на первом уровне поддержки (диспетчеры), в этом же окне заполняются поля «Решение», «Затраченное время», «Сложность» и инцидент закрывается автоматически.

В окне регистрации инцидентов отражаются данные о дежурных специалистах и текущих заменах. При необходимости осуществляется контекстный поиск известной ошибки в базе знаний, используется возможность просмотра истории событий КЕ.

Для работы с зарегистрированным, но нерешенным инцидентом, необходимо перейти в раздел Инциденты->Изменение статуса.

В списке не закрытых инцидентов выбирается необходимый инцидент, после чего появляется окно редактирования инцидента.

В окне редактирования, помимо контекстного поиска решения, можно изменить данные пользователя.

Проблема, не связанная с инцидентами регистрируется по тем же шагам, что и инцидент. Для работы с зарегистрированной проблемой используется окно редактирования проблемы.

Если среди предложенных конфигураций не оказалось подходящей, в окне регистрации конфигураций можно создать новую (рисунок 6.5).

					<i>27.03.04.2020.046.00.00 ПЗ ВКР</i>	<i>Лист</i>
<i>Изм.</i>	<i>Лист</i>	<i>№ докум.</i>	<i>Подпись</i>	<i>Дата</i>		57

В разделе «Справка» размещены ссылки на методические материалы и документацию.

Безопасность системы обеспечивается за счет разграничения прав доступа, которые регламентируют доступные группы КЕ и разделы программы.

Сотрудники обеспечивают безопасность доступа посредством аутентификации по имени и паролю. Для шифрования (преобразование в форму, в которой информацию трудно распознать) паролей пользователей данной системы используется функция PHP crypt().

Функция crypt — является удобным средством для одностороннего шифрования данных. Под «односторонним шифрованием» понимается, что данные могут только шифроваться – алгоритмы для расшифровки данных, обработанных функцией crypt(), пока неизвестны.[13]

Структура программных файлов представлена на рисунке 6.8.

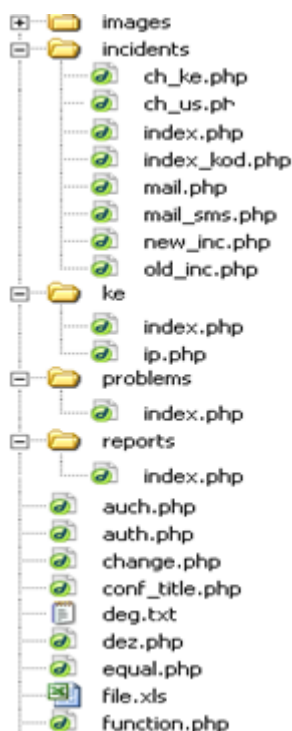


Рисунок 6.8 – Файловая структура проекта

Общие, наиболее часто используемые функции хранятся в файле «function.php». Кроме того, в файле «library.php» хранятся переменные, используемые всеми модулями программы. Файлы «function.php» и «library.php» включаются во все основные файлы программы. Для формирования единого шаблона страниц используются корневой файл «index.php» (включающий «menu.php») и «indexend.php». Все основные программные файлы включают скрипт проверки прав доступа, который хранится в файле «auch.php».

Файл «function.php» имеет следующий вид:

```
<?
// соединение с базой ORACLE
$ch="servdesc";
$tns="sp4db2";
$login_db="zav";
$password="*****";
$c1=OCILogon($login_db,$password,$tns) or die("Внимание! В данный момент соединение с базой прервано.");
```

Пользователю либо открывается окно ввода логина и пароля, либо выводится сообщение "Внимание! В данный момент соединение с базой прервано."

```
// Функция для определения выпадающего списка
function select_data_spisok($conn,$table,$Val1,$Val2)
{
    global $a; $a=array("");
    $stmt = OCIParse($conn,"select ".$Val1." from ".$table." ".$Val2);
    OCIExecute($stmt); OCICommit($conn);
    while ( OCIFetch($stmt) )
    {
        $cv = OCIResult($stmt,1);
```

					27.03.04.2020.046.00.00 ПЗ ВКР	Лист
Изм.	Лист	№ докум.	Подпись	Дата		60

```

// функция добавления нескольких элементов в конец массива
    array_push($a,"$cv");
}
OCIFreeStatement($stmt);
}

// Функция ввода данных в базу
function insert_data($conn,$table,$Val1,$Val2)
{
    $stmt=OCIParse($conn,"insert into ".$table." ".$Val1." values ".$Val2);
    OCIExecute($stmt); OCICommit($conn);
}

// Функция считывания поля в переменную $Vibor
function select_Vibor($conn,$table,$Val1,$Val2)
{
    global $Vibor;
    $stmt = OCIParse($conn,"select ".$Val1." from ".$table." ".$Val2);
    OCIExecute($stmt); OCICommit($conn);
    while ( OCIFetch($stmt) )
    {
        $Vibor = OCIResult($stmt,1);
    }
    OCIFreeStatement($stmt);
}

// Функция вывода данных в виде таблицы
function select_data2($conn,$table,$col_name,$ncol,$Val1,$Val2, $arr)
{
    global $menu, $id_ke, $res, $radio_error, $find_solution, $find_problem;
    $err_category=array("пусто", "программное обеспечение", "аппаратные
    средства", "периферия", "другое");

```

```

$stmt = OCIParse($conn,"select ".$Val1." from ".$table." ".$Val2);
OCIExecute($stmt); OCICommit($conn);
$j=1;
while ( OCIFetch($stmt) )
{
if ($j==1)
{
print"<table cellpadding=2 cellspacing=0 border=1 align=center width=98%><tr>";
for ($i = 0; $i <= $ncol-1; $i++)
{ print "<td class=textbb1 align=center>".$col_name[$i]."</td>"; }
print"</tr>";
$j++;
}
for ($i = 1; $i <= $ncol; $i++)
{
$res=OCIResult($stmt,1);
$cv[$i] = OCIResult($stmt,$i);
$field_name=OCIColumnName($stmt,$i);
if (($menu=="old_prob" || $menu=="old_error" || $menu=="old_change")&&
$field_name=="STATUS")
{
$id=$ncol-($ncol-1);
if ($cv[$i] != "" and $cv[$i]!="закрыта")
{ print"<td class=textbn align=center><a href=index.php?check_status=true
&menu=".$menu."&id_ke=".$cv[$id].">".$cv[$i]."&nbsp;</a></td>"; }
elseif ($cv[$i] != "" and $cv[$i]=="закрыта")
{ print"<td class=textbn align=center>".$cv[$i]."&nbsp;</td>"; }
else
{ print"<td class=textbn align=center>&nbsp;</td>"; }
}
else
{

```

					<i>27.03.04.2020.046.00.00 ПЗ ВКР</i>	<i>Лист</i>
<i>Изм.</i>	<i>Лист</i>	<i>№ докум.</i>	<i>Подпись</i>	<i>Дата</i>		<i>62</i>

```

if ($cv[$i] != "")
{
    $x=$cv[$i];
    if ($field_name=="CHANGE_FIELD_NAME" or $field_name=="ID_KE_TAB"
        or $field_name=="PRIORITY")
    { print"<td class=txtbn align=center>".$arr[$x]."&nbsp;</td>"; }
    elseif ($field_name=="CATEGORY")
    { print"<td class=txtbn align=center>".$err_category[$x]."&nbsp;</td>"; }
    else
    {
        if ($field_name=="ID" and ($find_problem or $find_solution))
        { print"<td class=txtbn align=center><input type=radio value=\"\". $cv[$i].\" \"
            name=radio_error></td>"; }
        else { print"<td class=txtbn align=center>".$cv[$i]."&nbsp;</td>"; }
    }
}
else
{ print"<td class=txtbn align=center>&nbsp;</td>"; }
}
}
print"</tr>";
}
if ($j==2) {print"</table>";}
OCIFreeStatement($stmt);
}

```

// Функция вывода данных в виде таблицы для инцидентов

```
function select_data3($conn,$table,$col_name,$ncol,$Val1,$Val2)
```

```

{
    global $menu, $id_ke, $id_problems;
    $stmt = OCIParse($conn,"select ".$Val1." from ".$table." ".$Val2);
    OCIExecute($stmt); OCICommit($conn);
}

```

					<i>27.03.04.2020.046.00.00 ПЗ ВКР</i>	<i>Лист</i>
<i>Изм.</i>	<i>Лист</i>	<i>№ докум.</i>	<i>Подпись</i>	<i>Дата</i>		<i>63</i>

```

print"<table cellpadding=2 cellspacing=0 border=1 align=center width=95%><tr>";
$ncol1=$ncol-2;
for ($i = 0; $i <=$ncol1; $i++)
{ print "<td class=textb_r align=center>".$scol_name[$i]."</td>"; }
print"</tr>";

while ( OCIFetch($stmt) )
{
print"<tr>";
$colval = OCIResult($stmt,1);
for ($i = 1; $i <= $ncol; $i++)
{
$cv[$i] = OCIResult($stmt,$i); $field_name=OCIColumnName($stmt,$i);
if ($cv[$i] != "")
{
print"<td class=textbn align=center>".$cv[$i]."&nbsp;</td>";
}
else
{
print"<td class=textbn align=center>&nbsp;</td>";
}
}
print"</tr>";
}
print"</table>";
OCIFreeStatement($stmt);
}

```

Основной файл работы с инцидентами приведен в приложении А.

					<i>27.03.04.2020.046.00.00 ПЗ ВКР</i>	<i>Лист</i>
<i>Изм.</i>	<i>Лист</i>	<i>№ докум.</i>	<i>Подпись</i>	<i>Дата</i>		64

ЗАКЛЮЧЕНИЕ

В ходе выполнения данной выпускной квалификационной работы выполнены следующие действия:

- проведён анализ технического задания;
- проведено исследование алгоритма работы системы;
- разработаны алгоритм работы поддержки пользователей информационных сервисов и схема базы данных;
- разработано программное обеспечение для отслеживания эффективной работы ИТ-специалистов в режиме реального времени на основе web-технологий. В разработанном ПО все сведения о сбоях, изменениях регистрируются и хранятся в единой базе данных; пользователи получили возможность контролировать ход восстановительных работ, получая автоматизированные почтовые уведомления о смене статуса их запроса, ответственном исполнителе, предпринятых действиях; специалисты получили доступ к постоянно накапливающимся сведениям о произошедших сбоях, проведенных изменениях в привязке к определенной конфигурационной единице, а также известным ошибкам.

Требования технического задания выполнены в полном объёме, разработка завершена.

					<i>27.03.04.2020.046.00.00 ПЗ ВКР</i>	<i>Лист</i>
<i>Изм.</i>	<i>Лист</i>	<i>№ докум.</i>	<i>Подпись</i>	<i>Дата</i>		<i>65</i>

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1 Кайт, Т. Oracle для профессионалов. Технологии и решения для достижения высокой производительности и эффективности / Томас Кайт , Дарл Кун. - М.: Вильямс, 2015. - 960 с.

2 Браун, Брэдли Oracle Database. Создание Web-приложений / Брэдли Браун , Ричард Дж. Ниемик, Джозеф С. Треззо. - М.: ЛОРИ, 2010. - 722 с.

3 Дунаев, В. Сценарии для Web-сайта. PHP и JavaScript / В. Дунаев. - М.: БХВ-Петербург, 2017. - 576 с.

4 Дронов, Владимир PHP 5/6, MySQL 5/6 и Dreamweaver CS4. Разработка интерактивных Web-сайтов / Владимир Дронов. - М.: БХВ-Петербург, 2017. - 544 с.

5 Гвоздева, В.А. Информатика, автоматизированные информационные технологии и системы: Учебник (ГРИФ) / В.А. Гвоздева. - М.: Форум, 2017. - 544 с.

6 Информационные системы и технологии / Под ред. Тельнова Ю.Ф.. - М.: Юнити, 2017. - 544 с.

7 Бородакий, Ю.В. Информационные технологии. Методы, процессы, системы / Ю.В. Бородакий, Ю.Г. Лободинский. - М.: ГЛТ , 2014. - 456 с.

8 Хардман, Рон Oracle Database PL/SQL. Рекомендации эксперта / Рон Хардман , Майкл МакЛафлин. - Москва: Лори, 2014. - 450 с.

9 Ленгсторф, Джейсон PHP и jQuery для профессионалов / Джейсон Ленгсторф. - М.: Вильямс, 2017. - 362 с.

10 Бенкен, Е. PHP, MySQL, XML. Программирование для Интернета / Е. Бенкен. - М.: БХВ-Петербург, 2016. - 352 с.

11 Фленов, М.Е. PHP глазами хакера / М.Е. Фленов. - 2-е изд., доп. и перераб. - Санкт-Петербург : БХВ-Петербург, 2011. - 330 с.

					27.03.04.2020.046.00.00 ПЗ ВКР	Лист
Изм.	Лист	№ докум.	Подпись	Дата		66

12 Варфоломеева, А.О. Информационные системы предприятия: Учебное пособие / А.О. Варфоломеева, А.В. Коряковский, В.П. Романов. - М.: НИЦ Инфра-М, 2015. - 283 с.

13 Кайт, Т. Oracle для профессионалов. Архитектура, методики программирования и особенности версий 9i, 10g и 11g / Т. Кайт. - М.: Диалектика / Вильямс, 2013. - 119 с.

14 <http://www.i-teco.ru/article3.html>

15 <http://www.itexpert.ru/rus/biblio/itil/>

16 www.itsm.ru

17 <http://ru.wikipedia.org/wiki/ITSM>

18 <http://www.itexpert.ru/rus/biblio/itil/> презентация Основы ITSM

19 <http://www.cisp.org.ua/cisp/news.nsf/OpenDocument>

					<i>27.03.04.2020.046.00.00 ПЗ ВКР</i>	<i>Лист</i>
<i>Изм.</i>	<i>Лист</i>	<i>№ докум.</i>	<i>Подпись</i>	<i>Дата</i>		<i>67</i>

ПРИЛОЖЕНИЕ А

Исходный код ПО технической поддержки пользователей информационных сервисов

```
// Основной файл для работы с инцидентами //
<?
session_start();

// доступ по правам пользователя

if ($prava=='1' or $prava=='2' or $prava=='5' or $prava=='4' or
$prava=='7') {

    include("../index.php"); ?>
<table width="100%">
<tr>
<td bgcolor="2E4280" class=twb2 align=center>
<? print"$title"; ?>
</td>
</tr>
</table>
<table width="95%">
<tr>
<td align=left valign=top><?
// вывод дежурных ИТ-специалистов
if ($ndeg) {
$file_array=file("../deg.txt");

if ($worker1 == "") { $worker1=$file_array[0]; }
if ($worker2 == "") { $worker2=$file_array[1]; }
if ($worker3 == "") { $worker3=$file_array[2]; }
if ($worker4 == "") { $worker4=$file_array[3]; }

$fh=fopen("../deg.txt","w+");
$success=fwrite($fh,$worker1);
$success = fwrite($fh,"\n");
$success=fwrite($fh,$worker2);
$success = fwrite($fh,"\n");
$success=fwrite($fh,$worker3);
```

					27.03.04.2020.046.00.00 ПЗ ВКР	Лист
Изм.	Лист	№ докум.	Подпись	Дата		68

```

    $success = fwrite($fh, "\n");
    $success=fwrite($fh,$worker4);
    fclose($fh);
}
$ww=file("../deg.txt");
</td><td valign=top align=left >";
    $stmt = OCIParse($c1,"select ".$sch." ".$stab17.".FIO,
    ".$sch." ".$stab17.".SMENA,
        ".$sch." ".$stab17.".S_OTSTUT, ".$sch." ".$stab17.".E_OTSTUT from
        ".$sch." ".$stab17." where ".$sch." ".$stab17.".OTSUTSTV='1'");
    OCIExecute($stmt);
    OCICommit($c1);
    print"<table cellpadding=3 cellspacing=0 border=0 align=left
class=textsb2><tr>";
        $arr=array ("Отсутствует", "Заменяет", "Время отсутствия");
    $s=sizeof($arr);
    for ($i = 0; $i <$s; $i++){
        print "<td align=left>".$arr[$i]."</td>";
    }
    print"</tr>";
    while ( OCIFetch($stmt) )
    {
        $cv1 = OCIResult($stmt,1);
        $cv2 = OCIResult($stmt,2);
        $cv3 = OCIResult($stmt,3);
        $cv4 = OCIResult($stmt,4);
        print "<tr><td>$cv1</td><td>$cv2</td><td>$cv3 -
    $cv4</td></tr>";
    }
    print"</table></td></tr></table>";
// функция include() подключает файл «new_inc.php»
    include("new_inc.php");
// функция include() подключает файл «old_inc.php»
    include("old_inc.php");

```

					27.03.04.2020.046.00.00 ПЗ ВКР	Лист
Изм	Лист	№ докум	Подпись	Дата		69

```

include("index_kod.php");
include("../indexend.php");
}
?>

// Окно «Регистрация инцидента»
//Скрипт запускается функцией include() в основном «файле index.php»
//*****
<?
// если выбраны опции «История КЕ» запускаем файл history.php
if ($show_history){
    include("../history.php");
}
//если выбран пункт меню «Регистрация инцидента» и не выбрана «История
КЕ»
if ($menu=="new_inc" AND !$show_history) { ?>
    <form action="index.php?menu=<? print $menu; ?>" method="post"
name="form1">
        <input type="hidden" name="menu" value="<? print $menu; ?>">
        <input type="hidden" name="id_ke_tab" value="<? print $id_ke_tab;
?>">
        <input type="hidden" name="id_ke" value="<? if (!$get_rm) { print
$id_ke; } ?>">
        <input type="hidden" name="get_rm" value="<? print $get_rm; ?>">
        <input type="hidden" name="rm" value="<? print $rm; ?>">
        <? if (!$record) { ?>
            <table width=98% align=center cellpadding=5 cellspacing=5
style="background:
                #F8F8F8; border: thin dotted #EDED; border="0">
            <tr>
                <td class="textbb" width=30%>Тип конфигурационной единицы:<br>
                <? if ($id_ke_tab=="KE_ACCOUNT") {
                    $ke_tab="account-сервис";

```

					27.03.04.2020.046.00.00 ПЗ ВКР	Лист
Изм.	Лист	№ докум.	Подпись	Дата		70

Продолжение приложения А

```

}
if ($id_ke_tab=="KE_HARD") {
    $ke_tab="аппаратные средства";
}
if ($id_ke_tab=="KE_PERIPHERY") {
    $ke_tab="периферия";
}
if ($id_ke_tab=="KE_SOFT") {
    $ke_tab="программное обеспечение";
} ?>
<select name="id_ke_tab" class=textbtn onchange="if
    (this.options[this.selectedIndex].value=='KE_HARD' ||
    this.options[this.selectedIndex].value=='KE_PERIPHERY')
    {t.innerHTML='Инвентарный номер';
f_inv.style.visibility='visible';
    f_inv.value=''} elseif
    (this.options[this.selectedIndex].value=='KE_ACCOUNT')
    {t.innerHTML='Account' ; f_inv.style.visibility='visible';
f_inv.value=''}
    else {t.innerHTML=''; f_inv.style.visibility='hidden';
f_inv.value=''};">
    <? print"<option value =
    '". $id_ke_tab. "'>". $ke_tab. "</option>"; ?>
        <option value = "KE_ACCOUNT">account-
сервис</option>
        <option value = "KE_HARD">аппаратное
средство</option>
        <option value =
"KE_PERIPHERY">периферия</option>
        <option value = "KE_SOFT">программное
обеспечение</option>
    </select>
<br>Постоянный номер:<br>
    <input name="rm" type="text" size="10" maxlength="6"
class="textbtn"
    value="<? print $rm; ?>"><br>

```

					27.03.04.2020.046.00.00 ПЗ ВКР	Лист
Изм.	Лист	№ докум.	Подпись	Дата		71

Продолжение приложения А

```

<div id="t" class="textbb">
    <? if ($sid_ke_tab=="KE_PERIPHERY" or $sid_ke_tab=="KE_HARD"
)
        {
            print "Инвентарный номер";
        }
        elseif ($sid_ke_tab=="KE_ACCOUNT" ) {
            print"Account";
        }?>
</div>
<input name="f_inv" type="text" size="15" maxlength="15"
class="textbn"
    value="<? print $f_inv; ?>" style="visibility:<? if
($sid_ke_tab and
    ($sid_ke_tab=="KE_ACCOUNT" or $sid_ke_tab=="KE_PERIPHERY" or
    $sid_ke_tab=="KE_HARD")){print "visible";} else
{print"hidden";}?>"><br>
    <input name="get_rm" type="submit" value="Выбрать KE"
class="textbb">
</td>
<td class="textbb" width=30% valign="top">
    Фамилия:<br><input name="surname" type="text" size="20"
maxlength="50"
    class="textbn" value="<? print $surname; ?>"><br>
    Имя:<br><input name="name" type="text" size="20"
maxlength="50"
    class="textbn" value="<? print $name; ?>"><br>
    Отчество:<br><input name="patronymic" type="text" size="20"
maxlength="50" class="textbn" value="<? print $patronymic;
?>"><br>
</td>
<td align="right">
    <table width="80%">
        <tr>
            <td><input name="check_history" type="checkbox"
value="INCIDENTS">

```

					27.03.04.2020.046.00.00 ПЗ ВКР	Лист
Изм.	Лист	№ докум.	Подпись	Дата		72

Продолжение приложения А

```
Инциденты <br>
<input name="check_history1" type="checkbox"
value="PROBLEMS">

Проблемы <br>
<input name="check_history2" type="checkbox"
value="ERRORS">

Известные ошибки <br>
<input name="check_history3" type="checkbox"
value="CHANGES">

Изменения<br>
<input name="show_history" type="submit"
value="История КЕ" class="textbb" <? if
(!$id_ke) {print"disabled=true";}?>>
</td>
</tr>
</table>
</td>
</tr>
</table>
<? }
if (!$inc_insert) {
// запуск файла для выбора найденной КЕ
include("../get_rm.php");
// если найдена и выбрана КЕ выводим поля для записи инцидента
if (($get_rm and $id_ke_tab!="KE_SOFT") or $id_ke_hard) or
$find_solution) { ?>
<table width="90%" border="0" cellspacing="0" cellpadding="0"
align="center">
<tr valign="top">
<td width="35%" class=textbb>
Описание инцидента* (до 500 символов):<br>
<textarea name="trouble" class="textbn" cols="50"
rows="3">
<? print $trouble; ?>
</textarea><br>
```

Продолжение приложения А

Место расположения (до 500 символов):


```
<textarea name="way" class="textbn" cols="50" rows="3">
  <? print $way; ?>
</textarea><br>
```

* - обязательные для заполнения поля

</td>

<td width="30%" class=textbb>

Приоритет*:


```
<? print"<select name=priority class=textbn>
  <option value =
  '". $priority. "'>". $priority1. "</option>
  <option value = '3'>в порядке очереди</option>
  <option value = '2'>важно</option>
  <option value = '1'>очень
важно</option></select><br>"; ?>
```

Владелец:


```
<select name=ui_owner class="textbn">
  <? print"<option value =
  '". $ui_owner. "'>". $ui_owner. "</option>";
  $owner_usl="";
  select_data_spisok($c1,"$ch.$stab17","FIO","where
$ch.$stab17.UVOLEN
is null");
sort($a); $y=1;
while($y<sizeof($a)){
  select_Vibor($c1,"$ch.$stab17","
$ch.$stab17.OTSUTSTV "," where
$ch.$stab17.FIO='$a[$y]'""); $o=$Vibor; unset($Vibor);
  if ($o=='1'){
    select_Vibor($c1,"$ch.$stab17","
$ch.$stab17.S_OTSUT "," where
$ch.$stab17.FIO='$a[$y]'""); $s=$Vibor; unset($Vibor);
    select_Vibor($c1,"$ch.$stab17","
$ch.$stab17.E_OTSUT "," where
```

27.03.04.2020.046.00.00 ПЗ ВКР

Лист

74

Изм	Лист	№ докум	Подпись	Дата
-----	------	---------	---------	------

Продолжение приложения А

```

$ch.$stab17.FIO='$a[$y]'); $e=$Vibor; unset($Vibor);

        $s=substr_replace ($s, "20", 6,0);
        $s_arr=explode (".", $s);
        $e=substr_replace ($e, "20", 6,0);
        $e_arr=explode (".", $e);
        $e_u = mktime(0,0,0,$e_arr[1],
$e_arr[0],$e_arr[2]);

        $s_u = mktime(0,0,0,$s_arr[1],
$s_arr[0],$s_arr[2]);

        $now=mktime
(0,0,0,date("m"),date("d"),date("Y"));
    }
    if ($o!='1' or ($o=='1' and ($now<$s_u or $now>$e_u)))
    {
        print "<option value =
'".$a[$y]."'>".$a[$y]."</option>";
    }
    $y++;
} ?>

</select><br><br>
Примечание (до 500 символов):<br>
<textarea name="note" class="textbn" cols="50"
rows="3">

    <? print $note; ?>
</textarea><br>
</td>

<td width="35%" class=textbb style="background: #F8F8F8;
border: thin

        dotted #EDED; ">

        Поиск решения<br>
        по краткому описанию ошибки:<br>
        <input name="short_descr" value="<? print
$short_descr; ?>" type="text"

        size="50" class="textbn"><br>

```

					<p style="font-size: 1.2em; margin: 0;">27.03.04.2020.046.00.00 ПЗ ВКР</p>	<p style="font-size: 0.8em; margin: 0;">Лист</p>
Изм	Лист	№ докум	Подпись	Дата		75

Продолжение приложения А

по подробному описанию ошибки:


```
<input name="error_descr" value="<? print
$error_descr; ?>" type="text"
        size="50" class="textbn"><br>
<input name="find_solution" type="submit"
value="Искать решение"
        class="textbb"><br>
<? if ($find_solution and !$id_ke) {
        print"<p class=textbbred>Не выбрана
конфигурационная
        единица!</p>";
    } ?>
</td>
</tr>
<? if ($find_solution and $id_ke)
    { ?>
<tr>
    <td colspan="3" class=textbb style="background: #F8F8F8;
border: thin
        dotted #EDEDDED;">
        <? print"Возможные решения инцидента:";
        $zapros="where ".$sch.". ".$stab15.".SOLUTION is not
null";
            if ($short_descr) {
                $zapros=$zapros." and
".$sch.". ".$stab15.".SHORT_DESCR like
                    '%".$short_descr.%'";
            }
            if ($error_descr) {
                $zapros=$zapros." and
".$sch.". ".$stab15.".ERROR_DESCR like
                    '%".$error_descr.%'";
            }
            if ($ke) {
```

					27.03.04.2020.046.00.00 ПЗ ВКР	Лист
Изм	Лист	№ докум	Подпись	Дата		76

Продолжение приложения А

```

        $zapros=$zapros." and
". $ch.". ".$stab15.".ID_KE_TAB=
                                ' ".$sid_ke_tab."' and
". $ch.". ".$stab15.".ID_KE=' ".$sid_ke."'";
    }
    $sh1=array($field_names["ID"],
$field_names["SHORT_DESCR"],
$field_names["CATEGORY"],$field_names["SOLUTION"],
                                $field_names["REG_DATE"],
$field_names["NOTE"],
                                $field_names["STATUS"]);
    // ВЫВОД ТАБЛИЦЫ ПО ВОЗМОЖНОМУ РЕШЕНИЮ
    select_data2($c1,$ch.". ".$stab15,$sh1,7,$ch.". ".$stab15.".ID,
". $ch.". ".
    $stab15.".SHORT_DESCR,". $ch.". ".$stab15.".CATEGORY,". $ch.". ".
    $stab15.".SOLUTION,". $ch.". ".$stab15.".REG_DATE,
". $ch.". ".$stab15.".
    NOTE, ". $ch.". ".$stab15.".STATUS", $zapros." order by
". $ch.". ".
    $stab15.".REG_DATE, ". $ch.". ".$stab15.".CATEGORY",
$field_names);
    print"<font class=textbn>если вы отметите -
приоритет будет отдан
                                отмеченному возможному решению
инцидента</font>"; ?>
        </td>
    </tr><?
        } ?>
    <tr valign="top">
        <td width="40%" class=textbb>
            Решение: (до 500 символов):<br>
            <textarea name="diagn" class="textbn" cols="50"
rows="3">
                <? print $diagn; ?>
            </textarea>
        </td>

```

					27.03.04.2020.046.00.00 ПЗ ВКР	Лист
Изм	Лист	№ докум	Подпись	Дата		77

Продолжение приложения А

```

<td colspan=2 class=textbb>
    Затраченное время (в формате 0,2):<br>
    <input name="time" type="text" class="textbn"
size=10><br>
    Сложность:<br>
    <select name="difficulty" class="textbn">
        <option></option>
        <option value="3">легкая</option>
        <option value="2">средняя</option>
        <option value="1">сложная</option>
    </select>
</td>
</tr>
<tr valign="top">
    <td colspan=3 class=textbb>
        <input name="inc_insert" type="submit"
value="Регистрировать инцидент"
        class="textbb" onClick="return
confirm('Зарегистрировать инцидент?') ">
    </td>
</tr>
</table><?
}
}
// если нажата кнопка Регистрировать
if ($inc_insert and $id_ke_tab and $id_ke and $trouble and
$priority) {
    // Если не указан владелец, записывается тот, кто
регистрировал
    if ($ui_owner=="") {
        @select_Vibor($c1,"$ch.$stab17","FIO","where ID='$sdus'");
        $ui_owner=$Vibor; unset($Vibor);
    }
    // Если заполнено поле решение

```

Продолжение приложения А

```

if ($diagn) {
    $status_inc="закрыт";
}
// Если отмечена ошибка - записывается err"ID"
if ($radio_error) {
    $diagn="err".$radio_error;
    $status_inc="закрыт";
}
// Если не заполнено поле решение и не отмечена ошибка
if (!$radio_error and !$diagn) {
    unset($difficulty);
    unset($time);
    $status_inc="зарегистрирован";
}
// создаем новую запись об инциденте
insert_data($c1,"$ch.$tab1",
            "(ID, ID_KE_TAB, ID_KE, HARD_FIELD, TROUBLE,
PRIORITY,
            WAY, DATE_IN, TIME_IN, UI_OWNER,
            STATUS, DATE_OUT, TIME_OUT, TIME_UNIX, DIAGN,
            DIFFICULTY, TIME, REGISTRATOR, NOTE)",
            "($secl, '$id_ke_tab', '$id_ke', '$hard_field',
'$trouble', '$priority', '$way',
            '$d_run', '$t_run', '$ui_owner',
            '$status_inc', '$d_run', '$t_run', '$d_unix_run',
'$diagn', '$difficulty',
            '$time', '$registrator','$note')");
$sh2=array($field_names["ID"],$field_names["NAMEPODR"],
            $field_names["SURNAME"], $field_names["TROUBLE"],
            $field_names["WAY"], $field_names["UI_OWNER"],
            $field_names["PRIORITY"], $field_names["DATE_IN"],
            $field_names["DIAGN"], $field_names["DIFFICULTY"],
            $field_names["TIME"], $field_names["DATE_OUT"],
            $field_names["STATUS"], $field_names["REGISTRATOR"]);

```

					27.03.04.2020.046.00.00 ПЗ ВКР	Лист
Изм	Лист	№ докум	Подпись	Дата		79

Окончание приложения А

```
@select_Vibor($c1,"$ch.$id_ke_tab","ID_RM","where ID='$id_ke'");
    $idrm=$Vibor; unset($Vibor);
    // выводим запись о зарегистрированном инциденте

@select_data3($c1,"$ch.$stab1,$ch.$stab7,$ch.$stab19,$ch.$stab20",$sh2,18,
    "$ch.$stab1.ID,$ch.$stab20.NAMEPODR,$ch.$stab19.SURNAME,
    $ch.$stab19.NAME,$ch.$stab19.PATRONYMIC,
    $ch.$stab1.TROUBLE,$ch.$stab1.WAY,$ch.$stab1.UI_OWNER,
    $ch.$stab1.PRIORITY,$ch.$stab1.DATE_IN,$ch.$stab1.TIME_I
N,
    $ch.$stab1.DIAGN,$ch.$stab1.DIFFICULTY,$ch.$stab1.TIME,
    $ch.$stab1.DATE_OUT,$ch.$stab1.TIME_OUT,
    $ch.$stab1.STATUS,$ch.$stab1.REGISTRATOR",
        "where $ch.$stab7.ID='$idrm' and
    $ch.$stab7.CONST_NUMBER=$ch.$stab19.LIN and
        $ch.$stab19.XXX=$ch.$stab20.XXX and
        $ch.$stab1.TROUBLE='$trouble' and
    $ch.$stab1.ID_KE='$id_ke'");
    // подключаем файл отправки оповещения пользователям и ИТ-
специалистам
    include("mail.php");
}
?> </form><?
```

					27.03.04.2020.046.00.00 ПЗ ВКР	Лист
Изм.	Лист	№ докум.	Подпись	Дата		80