**РЕФЕРАТ**

Дипломный проект: 67 с., 28 рис., 19 табл., 19 источников, 6 прил.

WEB-ПРИЛОЖЕНИЕ, СИСТЕМА МОНИТОРИНГА, УСТРОЙСТВО, СЕРВИС, БАЗА ДАННЫХ, ПОРОГОВЫЕ ЗНАЧЕНИЯ, СОПРЯЖЁННЫЕ ГРАФИКИ, МОДУЛЬ.

Объектом разработки является система мониторинга активности устройств и сервисов локальной вычислительной сети.

Цель работы: разработка дополнительного функционала и вспомогательных модулей системы мониторинга активности устройств и сервисов локальной вычислительной сети.

В процессе проектирования выполнены следующие разработки:

* проведен анализ исходных данных к проекту;
* разработано техническое задание;
* проведен сравнительный анализ аналогов и прототипов;
* реализованы функциональные части и система в целом;
* рассчитана стоимость автоматизированной системы;
* описаны методики использования разработанной системы;

Область возможного практического применения: нефтяное месторождение.

Результатом явилась многофункциональная система мониторинга устройств локальной вычислительной сети, которая позволяет не только производить сбор данных, но и отображать их в различных формах (графики, текст, нотификации, предупреждения). Так же разработанная система позволяет конфигурировать устройства и сервисов, добавлять пороговые значения, отображать физическое и логическое положение устройств и сервисов. В разработанной системе предусмотрен экспорт данных в графические и текстовые форматы.

Разработанная система внедрена и на данный момент активно используется.

Автор работы подтверждает, что приведенный в дипломном проекте расчетно-аналитический материал объективно отражает состояние разрабатываемого объекта, все заимствованные из литературных и других источников теоретические и методологические положения и концепции сопровождаются ссылками на их авторов.

# Список условных сокращений

AJAX (Asynchronous Javascript and XML) асинхронный [JavaScript](http://ru.wikipedia.org/wiki/JavaScript) и [XML](http://ru.wikipedia.org/wiki/XML)

API (Application Programming Interface) интерфейс программирования приложений

CRUD Create Read Update Delete) создание, чтение обновление и удаление

[LAMP](http://ru.wikipedia.org/wiki/LAMP) Linux-Apache-MySQL-PHP

MVC (Model Ciew Controller) модель-представление-поведение

UML (Unified Modeling Language) унифицированный язык моделирования

[WAMP](http://ru.wikipedia.org/wiki/WAMP) Windows-Apache-MySQL-PHP

БД база данных

ЛВС локальная вычислительная сеть

ОС операционная система

СУБД система управления базами данных

# Содержание

[Введение 7](#_Toc389598003)

[1 Анализ исходных данных и разработка технического задания 9](#_Toc389598004)

[1.1 Сущность проблемы 9](#_Toc389598005)

[1.2 Анализ исходных требований 9](#_Toc389598006)

[1.3 Обзор и сравнительный анализ существующих программных продуктов 10](#_Toc389598007)

[1.4 Обзор и сравнительный анализ аналогов и прототипов   
разрабатываемых вспомогательных модулей 12](#_Toc389598008)

[1.5 Обоснование выбора фреймворка и субд 14](#_Toc389598009)

[1.6 Обоснование выбора среды разработки 15](#_Toc389598010)

[1.7 Разработка технического задания 17](#_Toc389598011)

[1.8 Постановка задачи проектирования 17](#_Toc389598012)

[2 Разработка системы мониторинга компьютерных систем и сетей 19](#_Toc389598013)

[2.1 Функциональная структура программного комплекса 19](#_Toc389598014)

[2.2 Варианты использования 19](#_Toc389598015)

[2.3 Информационное обеспечение проекта 20](#_Toc389598016)

[2.4 Разработка структуры базы данных 21](#_Toc389598017)

[2.5 Разработка пользовательского интерфейса 24](#_Toc389598018)

[3 Реализация и тестирование индивидуального задания 27](#_Toc389598019)

[3.1 Установка и настройка рабочей среды 27](#_Toc389598020)

[3.2 Реализация модуля ingraph 29](#_Toc389598021)

[3.3 Реализация модуля icinga 34](#_Toc389598022)

[3.4 Реализация модуля assetdb 37](#_Toc389598023)

[3.5 Тестирование программного продукта 39](#_Toc389598024)

[4 Экономическая часть 44](#_Toc389598025)

[4.1 Обоснование необходимости создания системы   
мониторинга компьютерных систем и сетей 44](#_Toc389598026)

[4.2 Этапы работ по созданию системы мониторинга   
компьютерных систем и сетей 44](#_Toc389598027)

[4.3 Расчет стоимости автоматизированной информационной системы 46](#_Toc389598028)

[4.4 Расчёт цены разрабатываемой системы мониторинга   
компьютерных систем и сетей 52](#_Toc389598029)

[4.5 Сравнительный технико-экономический анализ   
эффективности производства нового продукта 53](#_Toc389598030)

[5 Охрана труда 55](#_Toc389598031)

[5.1 Введение 55](#_Toc389598032)

[5.2 Промышленная санитария и гигиена труда 56](#_Toc389598033)

[5.3 Техническая безопасность 60](#_Toc389598034)

[5.4 Электробезопасность 61](#_Toc389598035)

[5.5 Пожарная безопасность 62](#_Toc389598036)

[Заключение 64](#_Toc389598037)

[Список используемых источников 65](#_Toc389598038)

Приложения 67

[Приложение А (обязательное) Техническое задание на разработку web- приложения](#_Toc389598039)

[Приложение Б (рекомендуемое) Диаграммы вариантов использования](#_Toc389598055)

[Приложение В (рекомендуемое) Схема базы данных repository](#_Toc389598056)

[Приложение Г (обязательное) Описание программы](#_Toc389598057)

[Приложение Д (обязательное) Руководство пользователя (оператора)](#_Toc389598059)

[Приложение Е (дополнительное) Пример экспорта данных](#_Toc389598060)

# Введение

В настоящее время автоматизация управления ресурсами становиться все более важным фактором успеха любого предприятия. Полноценная и эффективная система автоматизации – один из самых действенных способов снижения издержек и повышения доходов. Широчайшие возможности анализа и современные технологии автоматизации помогут качественно планировать работу, а также осуществлять мониторинг достигнутых целей. Автоматизация позволит на качественно ином уровне решать задачи управления бизнес-процессами, отношениями с клиентами и ценовой политикой, автоматизировать различные виды учета.

Существует много веских причин перевода существующей информации на компьютерную основу. Сейчас стоимость хранения информации в файлах ЭВМ дешевле, чем на бумаге. Базы данных позволяют хранить, структурировать информацию и извлекать оптимальным для пользователя образом. Использование клиент/серверных технологий позволяют сберечь значительные средства, а главное и время для получения необходимой информации, а также упрощают доступ и ведение, поскольку они основываются на комплексной обработке данных и централизации их хранения. Кроме того ЭВМ позволяет хранить любые форматы данных, текст, чертежи, данные в рукописной форме, фотографии, записи голоса и т.д.

Таким образом, быстрота и качество – вот те два фактора, определяющие успешный исход какой–либо коммерческой деятельности.

Одним из немаловажных аспектов развития компьютерных систем является их объединение в локально вычислительные сети для достижения большей вычислительной мощности. Однако здесь встаёт проблема управления и контроля объединённой системы. Управляемость сети подразумевает возможность централизованно контролировать состояние основных элементов сети, выявлять и разрешать проблемы, возникающие при работе сети, выполнять анализ производительности и планировать развитие сети. В идеале средства управления сетями представляют собой систему, осуществляющую наблюдение, контроль и управление каждым элементом сети – от простейших до самых сложных устройств, при этом такая система рассматривает сеть как единое целое, а не как разрозненный набор отдельных устройств. Управление конфигурацией обеспечивает то, что системные установки, опции или различные варианты версий использования системы являются целостными и соответствуют реальной ситуации. Примером в области ЛВС могут служить распространение ПО (программное обеспечение), контроль версий, управление пользовательскими установками.

Системы управления сетью (NetworkManagementSystems) – централизованные программные системы, которые собирают данные о состоянии узлов и коммуникационных устройств сети, а также данные о трафике, циркулирующем в сети. Эти системы не только осуществляют мониторинг и анализ сети, но и выполняют в автоматическом или полуавтоматическом режиме действия по управлению сетью - включение и отключение портов устройств, изменение параметров мостов адресных таблиц мостов, коммутаторов и маршрутизаторов и т.п. Примерами систем управления могут служить популярные системы HPOpenView, SunNetManager, IBMNetView.

Система мониторинга сети выполняет наблюдение за сетью в поисках проблем, вызванных перегруженными и/или отказавшими [серверами](http://ru.wikipedia.org/wiki/%D0%A1%D0%B5%D1%80%D0%B2%D0%B5%D1%80_(%D0%B0%D0%BF%D0%BF%D0%B0%D1%80%D0%B0%D1%82%D0%BD%D0%BE%D0%B5_%D0%BE%D0%B1%D0%B5%D1%81%D0%BF%D0%B5%D1%87%D0%B5%D0%BD%D0%B8%D0%B5)), другими устройствами или сетевыми соединениями.

Например, для того, чтобы определить состояние [web-сервера](http://ru.wikipedia.org/wiki/%D0%92%D0%B5%D0%B1-%D1%81%D0%B5%D1%80%D0%B2%D0%B5%D1%80" \o "Веб-сервер), программа, выполняющая мониторинг, может периодически отправлять запрос [HTTP](http://ru.wikipedia.org/wiki/HTTP) на получение страницы; для [почтовых серверов](http://ru.wikipedia.org/wiki/%D0%9F%D0%BE%D1%87%D1%82%D0%BE%D0%B2%D1%8B%D0%B9_%D1%81%D0%B5%D1%80%D0%B2%D0%B5%D1%80) можно отправить тестовое сообщение по [SMTP](http://ru.wikipedia.org/wiki/SMTP) и получить по [IMAP](http://ru.wikipedia.org/wiki/IMAP) или [POP3](http://ru.wikipedia.org/wiki/POP3).

Неудавшиеся запросы (например, в том случае, когда соединение не может быть установлено, оно завершается по [тайм-ауту](http://ru.wikipedia.org/w/index.php?title=%D0%A2%D0%B0%D0%B9%D0%BC-%D0%B0%D1%83%D1%82_(%D0%A2%D0%B5%D0%BB%D0%B5%D0%BA%D0%BE%D0%BC%D0%BC%D1%83%D0%BD%D0%B8%D0%BA%D0%B0%D1%86%D0%B8%D0%B8)&action=edit&redlink=1), или когда сообщение не было доставлено) обычно вызывают реакцию со стороны системы мониторинга. В качестве реакции может быть:

* отправлен сигнал тревоги [системному администратору](http://ru.wikipedia.org/wiki/%D0%A1%D0%B8%D1%81%D1%82%D0%B5%D0%BC%D0%BD%D1%8B%D0%B9_%D0%B0%D0%B4%D0%BC%D0%B8%D0%BD%D0%B8%D1%81%D1%82%D1%80%D0%B0%D1%82%D0%BE%D1%80);
* автоматически активирована система защиты от сбоев, которая временно выведет проблемный сервер из эксплуатации, до тех пор, пока проблема не будет решена;
* и так далее.

В рамках дипломного проектирования будет решаться проблема создания системы мониторинга и анализа данных активности устройств нефтяного месторождения.

Разрабатываемое программное обеспечение направлено на сокращение времени труда работников, занимающихся контролем работы оборудования на месторождение, быстрое оповещение операторов о неисправностях, формирование статистического анализа собранных данных об активности ЛВС.

# 1 Анализ исходных данных и разработка технического задания

## Сущность проблемы

Сущность проблемы заключается в необходимости реализации системы, которая позволила бы контролировать органы управления, компьютерные системы, датчики различных устройств на нефтяном месторождении.

Проблемы мониторинга компьютерных систем и сетей удобнее вести с использованием сетевых компьютерных средств, в частности web-сервера и web-приложения.

Мониторинг устройств и сервисов на этих устройствах должен вестись круглосуточно, так как малейший сбой в системе, который не будет своевременно предупреждён и устранён, может привести к катастрофическим последствиям, как экономическим, так и экологическим.

Нефтяное месторождение Салыма распространено на большой площади. По этому возникает проблема позиционирования контролируемых устройств.

Разрабатываемая система предназначена для администраторов и операторов устройств нефтяного месторождения, и как следствие возникает проблема в реализации и визуализации достаточного количества информации, для полного контроля устройств и сервисов. Визуальная информация должна иметь понятный для оператора язык, быть как можно более информативной и удобной в использовании и восприятии.

При отказе какого-либо устройства необходимо сразу же предупредить оператора и администратора системы, так как возникшая проблема может серьёзно повлиять на ход всего процесса нефтедобычи и нефтетранспортировки.

## Анализ исходных требований

Основные функции разрабатываемой системы:

* сбор данных об активности устройств и сервисов на данных устройствах;
* предоставление информации о данных в удобном и понятном виде;
* определение пороговых значений для собранных данных и определение статуса интервала, в который попадает собранное значение;
* сбор данных о сбоях в системе и своевременно информирование пользователя о нарушениях;
* реализация вспомогательного приложения для осуществления визуализации собранных данных в виде графиков;
* реализация вспомогательного приложения для управления устройствами, сервисами на этих устройствах, пороговыми значениями, группами устройств и сервисов и т.д.

Так же необходимо разработать базу данных для хранения информации о:

* пользователях, имеющих доступ к системе;
* всех устройствах;
* всех сервисах;
* взаимосвязи устройств и сервисов;
* местоположении устройств;
* логов активности на сайте;
* пороговых значениях для сервисов и устройств.

Разрабатываемая система должно иметь аутентификацию, гибкую систему перехода между модулями, удобный, понятный и настраиваемый интерфейс для каждого пользователя.

Любая информация, предоставляемая пользователю, должна иметь возможность экспорта в графический, текстовый и табличный форматы.

Для модуля управления устройствами необходимо разработать систему, позволяющую работать с БД, т.е. читать, изменять, удалять и добавлять данные, вводимые пользователем.

## Обзор и сравнительный анализ существующих программных продуктов

В качестве основного аналога разрабатываемой системы можно привести систему мониторинга ZabbiX.

ZABBIX – [свободная](http://ru.wikipedia.org/wiki/%D0%A1%D0%B2%D0%BE%D0%B1%D0%BE%D0%B4%D0%BD%D0%BE%D0%B5_%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%BD%D0%BE%D0%B5_%D0%BE%D0%B1%D0%B5%D1%81%D0%BF%D0%B5%D1%87%D0%B5%D0%BD%D0%B8%D0%B5) система мониторинга и отслеживания статусов разнообразных сервисов [компьютерной сети](http://ru.wikipedia.org/wiki/%D0%9A%D0%BE%D0%BC%D0%BF%D1%8C%D1%8E%D1%82%D0%B5%D1%80%D0%BD%D0%B0%D1%8F_%D1%81%D0%B5%D1%82%D1%8C), [серверов](http://ru.wikipedia.org/wiki/%D0%A1%D0%B5%D1%80%D0%B2%D0%B5%D1%80_(%D0%B0%D0%BF%D0%BF%D0%B0%D1%80%D0%B0%D1%82%D0%BD%D0%BE%D0%B5_%D0%BE%D0%B1%D0%B5%D1%81%D0%BF%D0%B5%D1%87%D0%B5%D0%BD%D0%B8%D0%B5)) и [сетевого оборудования](http://ru.wikipedia.org/wiki/%D0%A1%D0%B5%D1%82%D0%B5%D0%B2%D0%BE%D0%B5_%D0%BE%D0%B1%D0%BE%D1%80%D1%83%D0%B4%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5). Для хранения данных используется [MySQL](http://ru.wikipedia.org/wiki/MySQL" \o "MySQL), [PostgreSQL](http://ru.wikipedia.org/wiki/PostgreSQL" \o "PostgreSQL), [SQLite](http://ru.wikipedia.org/wiki/SQLite" \o "SQLite) или [Oracle](http://ru.wikipedia.org/wiki/Oracle_(%D0%A1%D0%A3%D0%91%D0%94)" \o "Oracle (СУБД)). Web-интерфейс написан на [PHP](http://ru.wikipedia.org/wiki/PHP). ZABBIX поддерживает несколько видов мониторинга:

* Simple checks – может проверять доступность и реакцию стандартных сервисов, таких как SMTP или HTTP без установки какого-либо программного обеспечения на наблюдаемом хосте;
* ZABBIX agent – может быть установлен на [UNIX-подобных](http://ru.wikipedia.org/wiki/UNIX) или [Windows](http://ru.wikipedia.org/wiki/Windows" \o "Windows) хостах для получения данных о нагрузке [процессора](http://ru.wikipedia.org/wiki/CPU), использования сети, дисковом пространстве и т.д.;
* External check – выполнение внешних программ. ZABBIX также поддерживает мониторинг через [SNMP](http://ru.wikipedia.org/wiki/Simple_Network_Management_Protocol).

Однако на месторождении использовалась устаревшая система Icinga, которую требовалось значительно доработать.

Icinga – система мониторинга,  представляющая собой ответвление системы мониторинга Nagios, [отделившийся](http://www.opennet.ru/opennews/art.shtml?num=28121) в начале мая 2009 года вследствие конфликта независимых разработчиков с компанией Nagios Enterprises. Исходные тексты Icinga распространяются в рамках лицензии GPL. Для загрузки подготовлены два архива - с классическим интерфейсом на языке С++ и новым интерфейсом на языке PHP. Дополнительно поставляется пакет Icinga-reporting, с реализацией аналитического модуля для генерации отчетов, написанного на языке Java и работающего вкупе с системой JasperServer.

От Nagios проект Icinga отличается переработанной архитектурой, переведенной на использование прослоек IDOMOD и IDO2DB, позволяющих организовать хранение данных мониторинга и конфигурационной информации в СУБД MySQL, Oracle или PostgreSQL. Структура Icinga оптимизирована для развертывания распределенных систем мониторинга, при которой возможно создание нескольких агентов мониторинга, осуществляющих проверки и направляющих результаты на основной узел. В состав Icinga входит более гибкий и удобный модульный web-интерфейс, переписанный на языке PHP, активно использующий AJAX и представляющий статистику в виде графиков. Для обеспечения интеграции с внешними сервисами предусмотрено несколько API: XML, JSON, SOAP.

На основе данной системы базируется мониторинг активности хостов Салымского нефтедобывающего комплекса. Пример данной системы можно увидеть на рисунке 1.1.

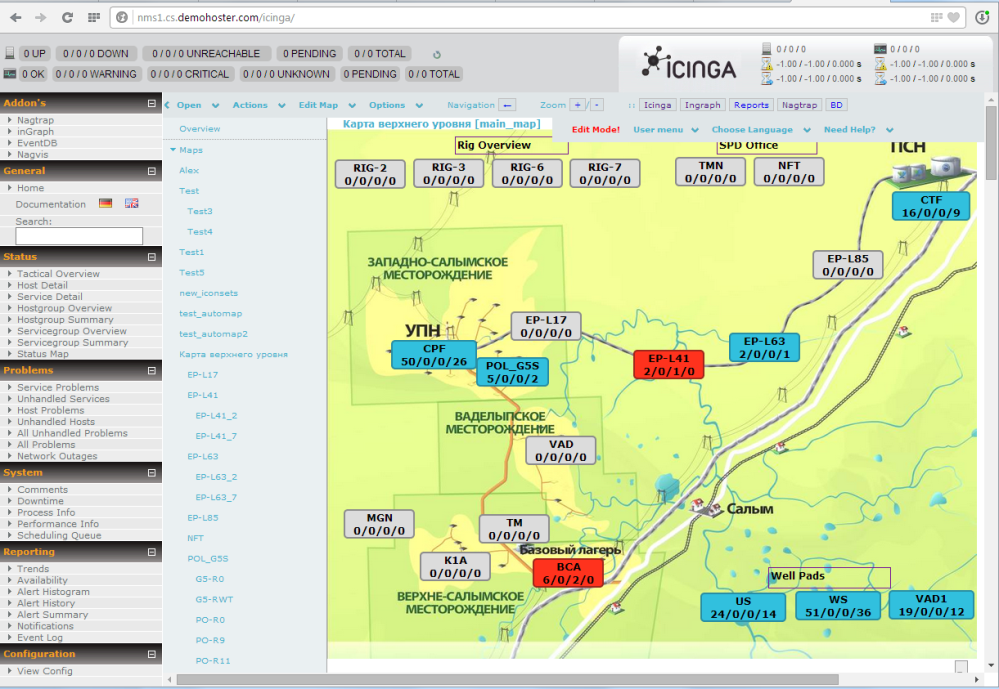


Рисунок 1.1 – Пример интерфейса системы мониторинга Icinga

Данная система уже запущенна и находится в эксплуатации. Задачей подразделения CS является усовершенствование данного продукта посредством реализации нового функционала.

Вспомогательный модуль управления не имеет аналогов и разрабатывается с нуля.

## Обзор и сравнительный анализ аналогов и прототипов разрабатываемых вспомогательных модулей

Рассмотрим системы отображения графической информации для разработки web-приложений. В качестве базового аналога рассмотрим библиотеку Flot. [Flot](http://code.google.com/p/flot/) – это JavaScript библиотека, предназначенная для рисования графиков. С ее помощью можно легко и быстро создавать красивые динамические графики, которые будут корректно работать в любом браузере независимо от того, какая ОС установлена у пользователя. В качестве ещё одного аналога можно привести библиотеку JavaScript Charts и PlotKit. Сравнительные характеристики данных библиотек по основным параметрам поставленной задачи предоставлены в таблице 1.1.

Таблица 1.1 – Основные характеристики графических библиотек

|  |  |  |  |
| --- | --- | --- | --- |
| **Характеристика** | **Flot** | **JavaScript Charts** | **PlotKit** |
| Поддержка IE 6+ | есть | нет | есть |
| Поддержка всех Linux, MacOS, Windows | есть | есть | нет |
| Привлекательность визуализации | высокая | высокая | низкая |
| Степень модульность | высокая | средняя | средняя |
| Степень интерактивности | высокая | высокая | средняя |
| Возможность масштабирования | есть | есть | есть |
| Степень простоты настройки отображения | высокая | средняя | средняя |
| Возможность вывода дополнительной текстовой информации | нет | нет | нет |
| Возможность отображения сопряжённого графика и анализ его сходимости | нет | нет | нет |

Основной задачей является взаимодействие скриптов PHP с базой данных MySQL. Как известно, расширения MySQL\_\* является устаревшим, начиная с версии PHP 5.5.0, и будет удалено в будущем. Соответственно необходимо использовать framework. Были рассмотрены framework Agavi+Doctrine и framework Yii.

Agavi – мощная, масштабируемая база PHP5 приложений, которая реализует парадигму Model-view-controller. Это позволяет разработчикам писать чистый, легко сопровождаемый и расширяемый код. Model-view-controller (MVC, «модель-представление-поведение», «модель-представление-контроллер», «модель-вид-контроллер») – схема использования нескольких [шаблонов проектирования](http://ru.wikipedia.org/wiki/%D0%A8%D0%B0%D0%B1%D0%BB%D0%BE%D0%BD_%D0%BF%D1%80%D0%BE%D0%B5%D0%BA%D1%82%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D1%8F), с помощью которых [модель данных](http://ru.wikipedia.org/wiki/%D0%9C%D0%BE%D0%B4%D0%B5%D0%BB%D1%8C_%D0%B4%D0%B0%D0%BD%D0%BD%D1%8B%D1%85) приложения, [пользовательский интерфейс](http://ru.wikipedia.org/wiki/%D0%9F%D0%BE%D0%BB%D1%8C%D0%B7%D0%BE%D0%B2%D0%B0%D1%82%D0%B5%D0%BB%D1%8C%D1%81%D0%BA%D0%B8%D0%B9_%D0%B8%D0%BD%D1%82%D0%B5%D1%80%D1%84%D0%B5%D0%B9%D1%81) и взаимодействие с пользователем разделены на три отдельных компонента таким образом, чтобы модификация одного из компонентов оказывала минимальное воздействие на остальные[15]. Схема данной модели представлена на рисунке 1.2.



Рисунок 1.2 – Концепция Model-View-Controller

Yii – это, как и Agavi, высокоэффективный основанный на компонентной структуре PHP-фреймворк для разработки масштабных web-приложений, поддерживающий структуру MVC. Он позволяет максимально применить концепцию повторного использования кода и может существенно ускорить процесс web-разработки. Название Yii означает простой (easy), эффективный (efficient) и расширяемый (extensible).

Doctrine 2 представляет собой хороший пример механизма объектно-реляционного отображения (ORM) для PHP 5.3+, позволяющий работать с базой данных максимально прозрачно, где в качестве промежуточного слоя используются обычные объекты PHP. В качестве основы используется мощный слой абстракции от базы данных (DBAL). Основная задача ORM – связать две концепции: объекты PHP и записи в реляционной базе данных[16].

Одна из ключевых особенностей Doctrine – возможность написания запросов на собственном объектно-ориентированном языке, чем-то напоминающим SQL, называемым Doctrine Query Language (DQL), созданным под вдохновением от Hibernates HQL. Помимо небольших отличий от SQL, он позволяет значительно усилить степень абстракции между объектами и строками базы данных, что позволяет создавать мощные и гибкие запросы, при этом сохраняя целостность [16].

Основные сравнительные характеристики Agavi и Yii приведены в таблице 1.2.

Таблица 1.2 – Основные характеристики каркасов web-приложений

|  |  |  |
| --- | --- | --- |
| **Характеристика** | **Agavi framework** | **Yii framework** |
| Язык | PHP+JS | PHP |
| Тип framework | [Web application framework](http://ru.wikipedia.org/wiki/Web_application_framework) | [Web application framework](http://ru.wikipedia.org/wiki/Web_application_framework) |
| Лицензия | [LGPL](http://ru.wikipedia.org/wiki/GNU_Lesser_General_Public_License) | [New BSD](http://ru.wikipedia.org/wiki/%D0%9B%D0%B8%D1%86%D0%B5%D0%BD%D0%B7%D0%B8%D1%8F_BSD) |
| Создание | 05.2005 | 01.2008 |
| MVC | да | да |
| Степень модульности | высокая | средняя |
| Скорость работы | высокая | очень высокая |
| Сложность | очень низкая | низкая |

## Обоснование выбора фреймворка и СУБД

На основе приведённых сравнительных характеристик framework и графических библиотек, было принято вести разработку с использованием Agavi framework и модуля flot для модуля графической визуализации информации по следующим причинам:

* низкая сложность изучения Agavi framework;
* бесплатная лицензия Agavi framework;
* парадигма MVC Agavi framework;
* высокая модульность Agavi framework;
* поддержка всех браузеров на всех OS библиотеки plot;
* высокая степень модульности библиотеки plot;
* гибкость настройки отображения графика с использованием библиотеки plot.

C использованием Yii Framework будет реализован модуль управления. Выбор данного фреймоворка обусловлен тем, что Yii идеально справляется с задачей управления БД. Но в отличии от Agavi не имеет функционала, позволяющего легко взаимодействовать с другими модулями.

В качестве СУБД было выбрано MySQL по причине того, что данная СУБД уже используется на месторождении и в ней находится значительный объём информации. MySQL – [свободная](http://ru.wikipedia.org/wiki/%D0%A1%D0%B2%D0%BE%D0%B1%D0%BE%D0%B4%D0%BD%D0%BE%D0%B5_%D0%9F%D0%9E) [реляционная система управления базами данных](http://ru.wikipedia.org/wiki/%D0%A0%D0%B5%D0%BB%D1%8F%D1%86%D0%B8%D0%BE%D0%BD%D0%BD%D0%B0%D1%8F_%D0%A1%D0%A3%D0%91%D0%94). Разработку и поддержку MySQL осуществляет корпорация [Oracle](http://ru.wikipedia.org/wiki/Oracle). MySQL является решением для малых и средних приложений. Входит в состав серверов [WAMP](http://ru.wikipedia.org/wiki/WAMP), [AppServ](http://ru.wikipedia.org/w/index.php?title=AppServ&action=edit&redlink=1), [LAMP](http://ru.wikipedia.org/wiki/LAMP). Данная СУБД легко настраивается и прекрасно взаимодействует с web-сервером Apache 2.2.

## Обоснование выбора среды разработки

Разработка ведётся на базе CentOs 6.5. Для более удобной работы, в качестве среды разработки было выбрано PHPStorm IDE. JetBrains PHPStorm – коммерческая кросс-платформенная [интегрированная среда разработки](http://ru.wikipedia.org/wiki/%D0%98%D0%BD%D1%82%D0%B5%D0%B3%D1%80%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%BD%D0%B0%D1%8F_%D1%81%D1%80%D0%B5%D0%B4%D0%B0_%D1%80%D0%B0%D0%B7%D1%80%D0%B0%D0%B1%D0%BE%D1%82%D0%BA%D0%B8) для [PHP](http://ru.wikipedia.org/wiki/PHP). Разрабатывается компанией [JetBrains](http://ru.wikipedia.org/wiki/JetBrains) на основе платформы [IntelliJ IDEA](http://ru.wikipedia.org/wiki/IntelliJ_IDEA).

PHPStorm представляет собой интеллектуальный редактор для [PHP](http://ru.wikipedia.org/wiki/PHP), [HTML](http://ru.wikipedia.org/wiki/HTML) и [JavaScript](http://ru.wikipedia.org/wiki/JavaScript" \o "JavaScript) с возможностями анализа кода на лету, предотвращения ошибок в коде и автоматизированными средствами [рефакторинга](http://ru.wikipedia.org/wiki/%D0%A0%D0%B5%D1%84%D0%B0%D0%BA%D1%82%D0%BE%D1%80%D0%B8%D0%BD%D0%B3" \o "Рефакторинг) для PHP и JavaScript. [Автодополнение](http://ru.wikipedia.org/wiki/%D0%90%D0%B2%D1%82%D0%BE%D0%B4%D0%BE%D0%BF%D0%BE%D0%BB%D0%BD%D0%B5%D0%BD%D0%B8%D0%B5" \o "Автодополнение) кода в PHPStorm поддерживает спецификацию PHP 5.3, 5.4 и 5.5 (современные и традиционные проекты), включая генераторы, сопрограммы, пространства имен, замыкания, типажи и синтаксис коротких массивов. Имеется полноценный [SQL](http://ru.wikipedia.org/wiki/SQL)-редактор с возможностью редактирования полученных результатов запросов. В рамках поставленной задачи PHPStorm является исключительно удобным средством так как:

* PHPStrom имеет возможность автозагрузки и синхронизации файлов и удалённым сервером;
* поддерживает синтаксис PHP, SQL, JavaScript;
* легко настраиваемый и расширяемый под решаемые задачи.

Разрабатываемая система базируется на концепции MVC, при этом клиентская часть (user interface) строится с использование JavaScript, в то время как бизнес логика работы с данными реализована на PHP.

В качестве тестируемой среды будет использоваться модуль ZendDebuger для PHP. Тестирование можно проводить с использованием следующих браузеров: Opera, FireFox, Chrome, Internet Explorer. Сравнительные характеристики браузеров приведены в таблице 1.3.

Таблица 1.3 – Сравнительные характеристики браузеров

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Характеристика** | **Opera** | **Fire Fox** | **Chrome** | **Internet Explorer** |
| Поддержка CSS2 | да | да | да | да |
| Поддержка JS | да | да | да | да |
| Поддержка HTML5 | да | да (с версии 4) | да | да (с версии 9) |
| Скорость работы c JS | средняя | средняя | высокая | низкая |
|  |  |  |  |  |
| Продолжение таблицы 1.3 | | | | |
| **Характеристика** | **Opera** | **Fire Fox** | **Chrome** | **Internet Explorer** |
| Скорость работы с CSS | высокая | средняя | средняя | низкая |
| Время загрузки | высокая | средняя | средняя | низкая |
| Использование памяти | средняя | низкая | низкая | высокая |
| Эмуляция версий | нет | нет | нет | есть |

Таким образом, исходя из анализа основных характеристик браузеров, приведенных в таблице 1.3, в качестве браузеров для тестирования были выбраны Opera (отладчик Dragonfly) и Internet Explorer. Браузер Opera будет использоваться для поверхностной отладки, так как её скорость работы высока и требуемые ресурсы незначительны. Удобный отладчик Dragonfly позволяет в интерактивном режиме наблюдать за процессами, проходящими в web-приложении (срабатывание js и jquery скриптов, отправка POST и GET запросов и т.д.). Internet Explorer предполагается использовать как браузер с минимальными характеристиками. Возможность эмуляции позволит отлаживать программы на версии IE 6, что гарантирует создание приложения, работающего на всех платформах.

## Разработка технического задания

Техническое задание необходимо для того, что бы определить основные исходные данные при проектировании информационной системы. Так же техническое задание необходимо для задания конкретных требований к системе и документации, условиям функционирования системы. На основании теоретического исследования, проведённого в рамках данной главы, были определены приоритетные факторы, влияющие на окончательный вид технического задания. Расширенное техническое задание на разрабатываемый программный продукт приведено в приложении А.

## Постановка задачи проектирования

Целью проектирования является разработка web-приложения, включающего в себя систему графического отображения данных, полученных приложением Icinga v1.10.0, разработку новой улучшенной версии системы мониторинга Icinga, разработка системы управления базой данных устройств и сервисов.

Разрабатываемое приложение должно быть доступным на любой платформе. Сайт, который необходимо реализовать должен иллюстрировать графики, отображающие данные по использованию сервисов на различных хостах. Система базируется на CentOS v6.5.

Сайт должен отображение комментария к графику на основании данных из базы данных Repository. Так же, необходимо на графиках выводить соответствующую информацию с сопряженных устройств (Input с устройства А должен будет совпадать с Output с устройства B).

В хранилище будут заданы связи между сервисами устройств. Если мы видим, что сервис, для которого отображается график, связан с другим сервисом, то:

* создается комментарий, который указывает, что данная связь присутствует;
* на экране отображается график связанного сервиса.

Необходимо реализовать базу данных repository, которая бы хранила данные о сервисах, хостах, сопряжённых сервисах (conjugate), а так же пороговых значениях сервисов (thresholds). Данная база данных должна быть связана с базой данных db.ingraph.

Данное программное обеспечение должно быть реализовано как отдельное web-приложение представленное в виде одного из модулей Icinga. Данные о хостах, сервисах и значения параметров, отображаемые на графиках, находятся в базе данных db.Icinga. Заполнение базы данных db.Icinga ведётся постоянно с помощью приложения Icinga.

Таким образом, разрабатываемое web-приложение должно выполнять следующие функции:

* считывать из базы данных список компьютерных систем и сервисов для них;
* предоставлять пользователю список компьютерных систем и сервисов, привязанных к ним, в режиме «живого поиска»;
* отображать информацию о компьютерных системах и сервисах в виде графика;
* отображать описание сервиса, если таковое существует;
* отображать информацию о сопряжённом устройстве и его сервисах, если таковое устройство существует;
* на графике должна быть предоставлена возможность выбора временного диапазона, осуществляться масштабирование;
* график должен иметь возможность экспорта в виде pdf или png файлов.

Так же необходимо расширить функционал библиотеки plot следующими модулями:

* модуль отображения сопряжённого графика;
* модуль отображения дополнительной информации;
* модуль подсветки точек графика и отображения их значений;
* модуль экспорта графика в pdf файл.

# 2 Разработка системы мониторинга компьютерных систем и сетей

## Функциональная структура программного комплекса

Реализуемой программное средство представляет собой систему, состоящую из нескольких модулей:

* Icinga;
* Icinga-web;
* Ingraph;
* AssetDB(Repository);
* NagiosQL;
* NagVis;
* NagTrap.

Однако NagiosQL, NagVis, NagTrap не входят в дипломное проектирование по причине того, что данные модули разрабатывались другой командой разработчиков.

Icinga – представляет собой модуль сбора информации и представления простейшего анализа полученных данных. Данная система реализована на языке С/С++.

Icinga-web – представляет собой web-оболочку для отображения более глубокого анализа собранных данных.

Ingraph – модуль для графического отображения информации. Данный модуль доступен из Icinga и Icinga-web как tooltip для сервисов.

AssetDB – модуль контроля и управления устройствами и сервисами, а также взаимодействием устройств и сервисов.

## Варианты использования

При создании диаграмм вариантов использования нужно выделить два вида сущностей – актеры и прецеденты. Актер – это множество логически связанных ролей, исполняемых при взаимодействии с прецедентами [19]. Прецедент – это описание множества последовательных событий, включая варианты, выполняемые системой, которые приводят к наблюдаемому актером результату [19].

Диаграмму вариантов использования разрабатываемой системы можно увидеть в приложении Б.

Для разрабатываемой системы мониторинга существует одна группа пользователей – администратор. На данном этапе ведётся разработка системы сквозной аутентификации.

## Информационное обеспечение проекта

В состав информационного обеспечения проекта на различных этапах разработки автоматизированной информационной системы коменданта учебных корпусов входят стандарты, которые приведены в таблице 2.1.

Таблица 2.1 – Информационное обеспечение

|  |  |  |
| --- | --- | --- |
| **Стандарты** | **Назначение** | **Этапы разработки** |
| UML | Язык для визуализации, специфицирования, конструирования и документирования систем | Проектирование |
| SQL | Структурированный язык запросов SQL (Structured Query Language) используется при создании запросов, а также для обновления и управления реляционными базами данных. | Проектирование Реализация |
| С | Язык программирования низкого уровня | Реализация |
| PHP | Язык программирования высокого уровня | Реализация |
| JavaScript | Язык программирования высокого уровня | Реализация |
| HTML | Язык разметки текста | Реализация |

Язык UML на этапе проектирования применялся для моделирования информационной системы при помощи CASE-средства Rational Rose Enterprise Architect 9.0.

Стандарт SQL применялся для реализации базы данных при помощи MySQL.

Язык программирования низкого уровня С применялся для разработки приложения мониторинга Icinga. Данное приложение представляет собой демон, который располагается на базе Unix подобных системах.

Языки программирования высокого уровня PHP и JavaScript применяется при реализации вспомогательных модулей Ingraph и AssetDB, а так же вспомогательного функционала для модуля Icinga.

Язык разметки гипертекста HTML применялся при создании пользовательского интерфейса.

## Разработка структуры базы данных

В данной главе представлена структура базы данных, согласно требованиям, выдвинутым заказчиком. База данных необходима для хранения вспомогательной информации о хостах и сервисах, пороговых значениях, сопряжённых сервисах, связи устройств и сервисов.

Сущности и связи базы данных приведённые в таблице 2.2.

Таблица 2.2 – Сущности и связи базы данных

|  |  |
| --- | --- |
| **Название сущности/связи** | **Описание** |
| contacts | Содержит информацию о контактах устройства |
| devices | Хранит информацию об устройстве |
| devices\_has\_services | Связь устройства и сервиса |
| owner | Хранит информацию о владельце устройства |
| relation\_devices\_services | Связь сопряжённых сервисов на устройствах |
| service\_thresholds | Хранит информацию о пороговых значениях сервиса |
| services | Хранит информацию о сервисах |
| ports | Хранит информацию о портах сервиса |
| status | Хранит информацию статусе активности хоста |

Сущность contacts представляет собой список контактов, включающих в себя email администратора и вспомогательную информацию о нём. Описание сущности представлено в таблице 2.3.

Таблица 2.3 – Сущность contacts

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Поле** | **Тип** | **Null** | **Ключ** | **По умолчанию** | **Дополнительно** |
| id\_contact | int(11) | No | Pri |  | AUTO\_INCREMENT |
| description\_contact | varchar(255) | Yes |  | NULL |  |

Сущность ownerпредставляет собой список владельцев устройства, включающих в себя информацию о владельце. Описание сущности представлено в таблице 2.4.

Таблица 2.4 – Сущность owner

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Поле** | **Тип** | **Null** | **Ключ** | **По умолчанию** | **Дополнительно** |
| id\_owner | int(11) | No | Pri |  | AUTO\_INCREMENT |
| description\_owner | varchar(255) | Yes |  | NULL |  |

Связь devices\_has\_services представляет собой список ключей, связывающих устройство и сервис. Данная связь представляет логическое расположение сервисов на устройствах. Описание связи представлено в таблице 2.5.

Таблица 2.5 – Связь devices\_has\_services

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Поле** | **Тип** | **Null** | **Ключ** | **По умолчанию** | **Дополнительно** |
| devices\_id\_device | int(11) | No | FK |  |  |
| services\_id\_service | int(11) | No | FK |  |  |
| has\_id | int(11) | No | Pri | NULL | AUTO\_INCREMENT |

Сущность devices представляет собой список устройств, включающих в себя информацию о устройствах, такую как название устройства, физический и логический адрес, тип, местоположение, владельца и т.д. Описание сущности представлено в таблице 2.6.

Таблица 2.6 – Сущность devices

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Поле** | **Тип** | **Null** | **Ключ** | **По умолчанию** | **Дополнительно** |
| id\_device | int(11) | No | Pri |  | AUTO\_INCREMENT |
| host\_name | varchar(50) | No |  |  |  |
| Mac | varchar(23) | No |  |  |  |
| Ip | varchar(40) | Yes |  | NULL |  |
| Type | int(11) | Yes |  | NULL |  |
| Location | varchar(255) | Yes |  | NULL |  |
| Owner | int(11) | Yes | FK | NULL |  |
| start\_date | timestamp | Yes |  | NULL |  |
| Status | int(11) | Yes |  | NULL |  |
| Uptime | int(11) | Yes |  | NULL |  |
| Contact | int(11) | Yes | FK | NULL |  |
| Model | varchar(50) | Yes |  | NULL |  |
| Serial | varchar(50) | Yes |  | NULL |  |
| Unauthorized | tinyint(1) | No |  | 1 |  |
| Archive | tinyint(1) | No |  | 0 |  |
| archiving\_date | tinyint(1) | No |  | 1 |  |

Связь relation\_devices\_services представляет собой список ключей, связывающих сопряжённые устройства. Данная связь представляет логическую сцепку входных и выходных устройств и сервисов. Описание связи представлено в таблице 2.7.

Таблица 2.7 – Связь relation\_devices\_services

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Поле** | **Тип** | **Null** | **Ключ** | **По умолчанию** | **Дополнительно** |
| Id | int(10) | No | PRI |  | AUTO\_INCREMENT |
| From | int(10) | No | FK |  |  |
| To | int(10) | No | FK |  |  |

Сущность ports представляет собой список открытых портов устройства, включающих в себя информацию о устройствах, номере порта и его приоритете. Описание сущности представлено в таблице 2.8.

Таблица 2.8 – Сущность ports

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Поле** | **Тип** | **Null** | **Ключ** | **По умолчанию** | **Дополнительно** |
| id\_port | int(11) | No | Pri |  | AUTO\_INCREMENT |
| port\_id\_device | int(11) | No | FK |  |  |
| port\_number | int(5) | No |  |  |  |
| port\_priority | tinyint(1) | No |  | 0 |  |

Связь thresholds представляет собой список пороговых значений, привязанных к какому-либо сервису. Описание связи представлено в таблице 2.9.

Таблица 2.9– Связь service\_thresholds

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Поле** | **Тип** | **Null** | **Ключ** | **По умолчанию** | **Дополнительно** |
| id\_service\_thresholds | int(11) | No | Pri |  | AUTO\_INCREMENT |
| id\_device | int(10) | No | FK |  |  |
| id\_service | int(10) | No | FK |  |  |
| Lbound | int(10) | Yes |  | NULL |  |
| Rbound | int(10) | Yes |  | NULL |  |
| Result | Enum  ('OK',  'WARNING',  'ERROR') | No |  |  |  |
| Sort | int(10) unsigned | No |  | 0 |  |

Сущность services представляет собой список доступных сервисов. Сущность включает в себя информацию о сервисе, его описание, порте, на котором работает сервис, передаваемую информацию и т.д. Описание сущности представлено в таблице 2.10.

Таблица 2.10 – Сущность services

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Поле** | **Тип** | **Null** | **Ключ** | **По умолчанию** | **Дополнительно** |
| id\_service | int(11) | No | Pri |  | AUTO\_INCREMENT |
| description\_service | varchar(255) | Yes |  | NULL |  |
| ingraph\_description | varchar(255) | Yes |  | NULL |  |
| Port | int(10) | Yes | FK | NULL |  |
| Use | varchar(255) | Yes |  | NULL |  |
| check\_command | varchar(255) | Yes |  | NULL |  |
| process\_perf\_data | tinyint(1) | Yes |  | NULL |  |
| Register | tinyint(1) | Yes |  | NULL |  |
| Thresholds | varchar(255) | Yes |  | NULL |  |

Сущность status представляет собой список доступных статусов сервиса. Описание сущности представлено в таблице 2.11.

Таблица 2.11 – Сущность status

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Поле** | **Тип** | **Null** | **Ключ** | **По умолчанию** | **Дополнительно** |
| id\_status | int(11) | No | Pri |  | AUTO\_INCREMENT |
| description\_status | varchar(255) | Yes |  | NULL |  |

Схема базы данных представлена в приложении В.

## Разработка пользовательского интерфейса

Немаловажной задачей является – разработка графического интерфейса пользователя. Согласно техническому заданию, программное обеспечение является кросс-платформенным web-приложением. Таким образом при разработки пользовательского интерфейса необходимо учесть поддержку всех видов и версий браузеров, начиная с IE 6.0.

Под графическим интерфейсом пользователя (Graphical User Interface – GUI) подразумевается тип экранного представления, при котором пользователь может выбирать команды, запускать задачи и просматривать списки файлов, указывая на пиктограммы или пункты в списках меню, показанных на экране. Действия могут, как правило, выполняться с помощью мыши, либо нажатием клавиш на клавиатуре.

Согласно требованиям заказчика структура пользовательского интерфейса модуля Ingrpah представлена на рисунке 2.1

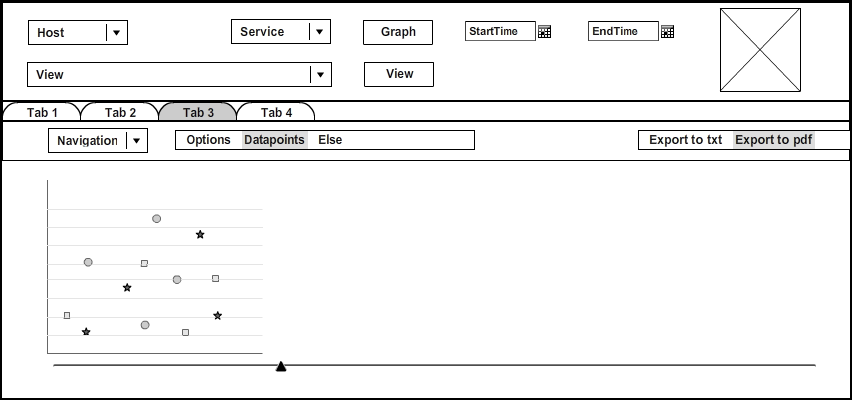


Рисунок 2.1 – Структура пользовательского интерфейса модуля Ingraph

Структура пользовательского интерфейса модуля Icinga представлена на рисунке 2.2.

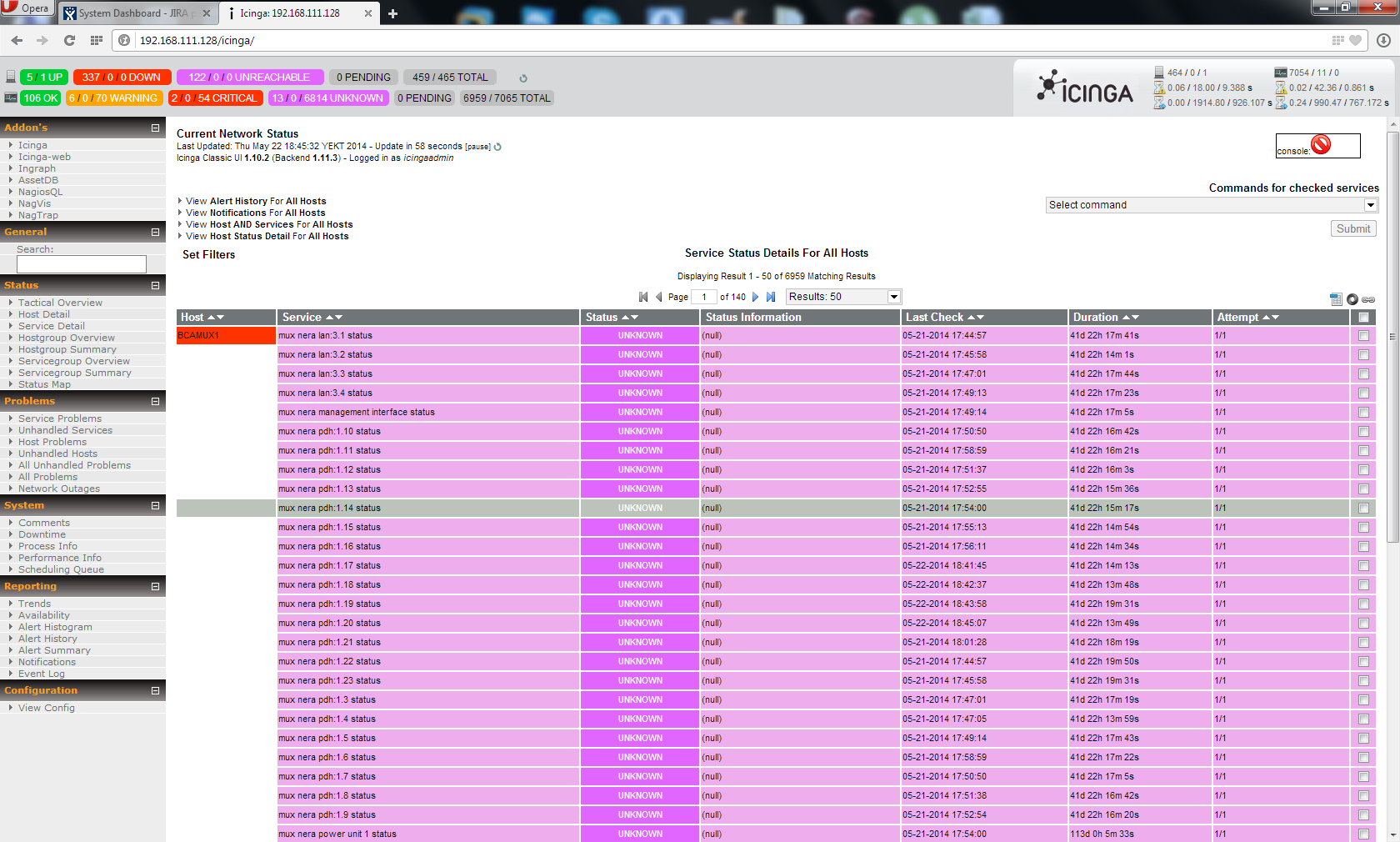


Рисунок 2.2 – Структура пользовательского интерфейса модуля Icinga

Структура пользовательского интерфейса модуля AssetDB представлена на рисунке 2.3.

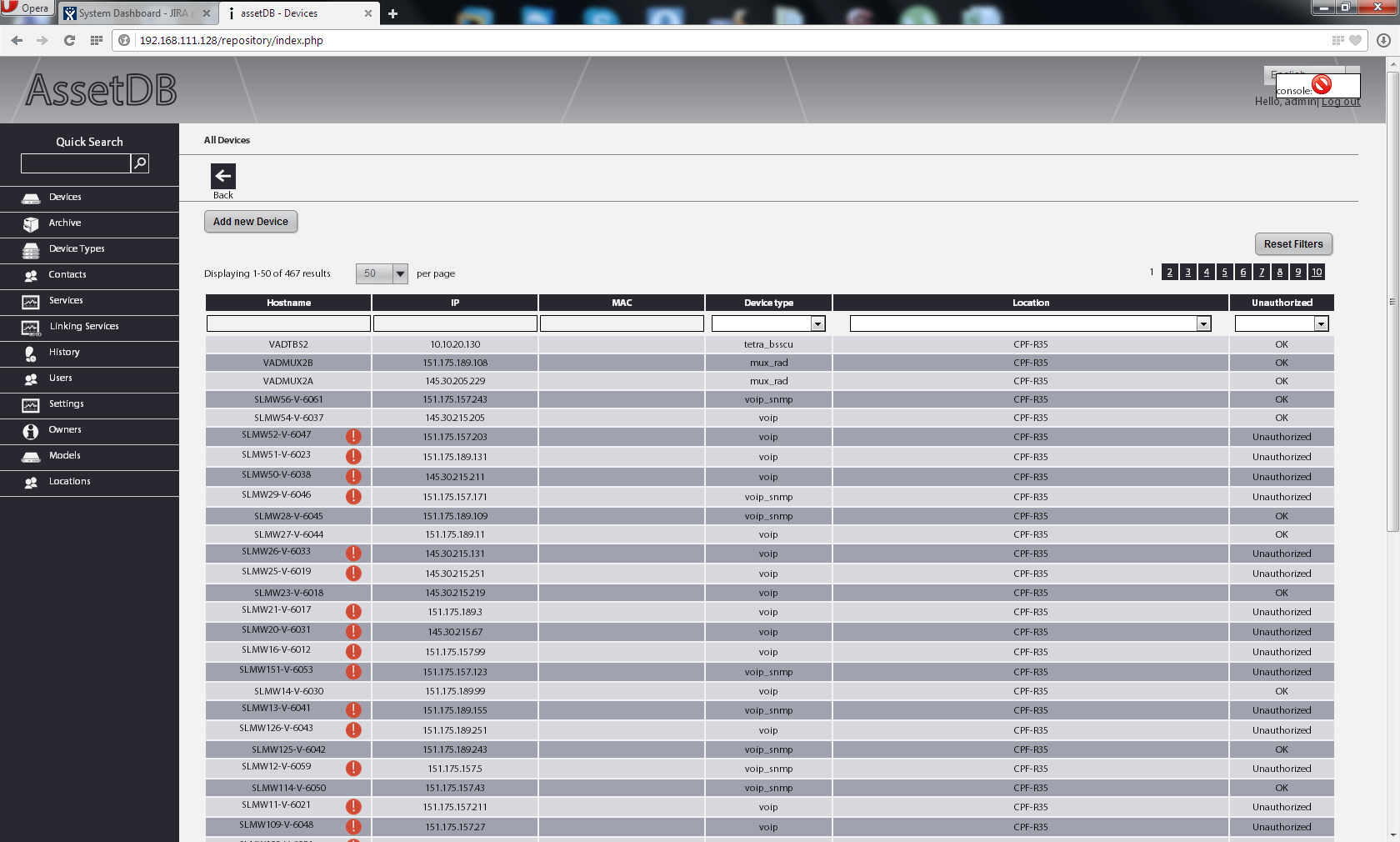


Рисунок 2.3 – Структура пользовательского интерфейса модуля AssetDB

# 3 Реализация и тестирование индивидуального задания

## Установка и настройка рабочей среды

Разрабатываемое приложение базируется на CentOS 6.0. После установки операционной системы на виртуальную машины для создания полноценного сервера необходимо установить Apache, PHP и MySQL. Все действия выполняются от пользователя root. Для установки системы необходимо выполнить следующие шаги.

Шаг 1. Необходимо обновить систему. Это делается следующей командой:

# yum update

Шаг 2. Для удобной работы необходимо установить Midnight Commander:

# yum -y install mc

Шаг 3. Установка Apache, SSL:

# yum -y install httpd mod\_ssl

Далее необходимо отредактировать конфигурационный файл apache. Для этого необходимо открыть в любимом редакторе файл **/etc/httpd/conf/httpd.conf.** В ServerName вносится адрес или имя сервера текущей копии системы. Прописывается Apache в автозапуск:

# chkconfig httpd on

Apache запускается командой:

# service httpd start

Для проверки работы сервера создаем тестовую страницу:

# echo '<h1>It Works!</h1>' > /var/www/html/index.html

Проверяем результат, открыв в браузере с другой машины IP-адрес сервера:

http://IP-адрес\_сервера/

Если надпись It Works! появилась, значит сервер работает корректно.

Шаг 4.  Установка PHP и его дополнительные компоненты. Для этого необходимо выполнить команду:

# yum -y install PHP PHP-common PHP-gd PHP-MySQL PHP-xml PHP-mbstring

Далее необходимо перезапустить Apache:

# service httpd restart

Проверить работу PHP можно выполнив следующий скрипт:

# echo '<?PHP PHPinfo(); ?>' > /var/www/html/inf.PHP

Результат работы данного скрипта проиллюстрирован на рисунке 3.1.

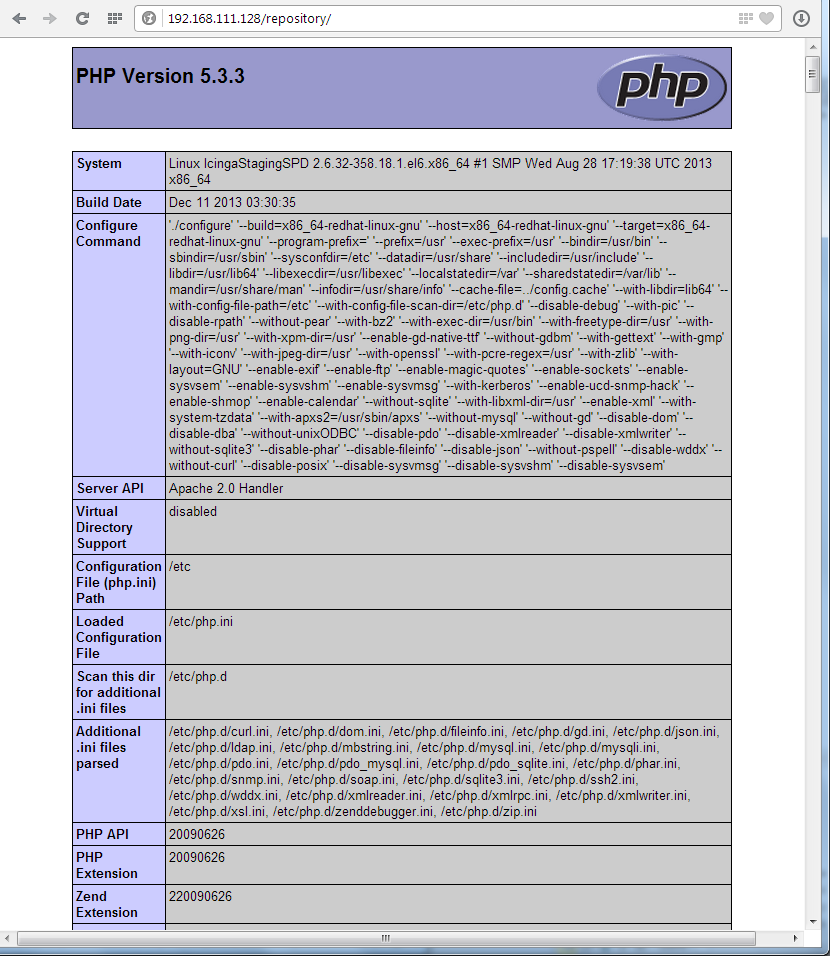


Рисунок 3.1 – Пример работы PHP

Шаг 5. Установка MySQL. Для установки MySQL выполняем команду:

# yum -y install MySQL MySQL-server

Добавляем MySQL в автозапуск командой:

# chkconfig MySQLd on

MySQL запускается следующей командой:

# service MySQLd start

Далее необходимо установить root пароль MySQL:

# MySQLadmin -u root password `root`

После выполнения данных операций, получается полноценный сервер на базе CentOS готовый к работе. Для удобной работы с платформой CentOS из Windows необходимо установить PHPStorm и настроить проект на автоматическую синхронизацию с сервером. Для этого необходимо открыть вкладку Tools/Deployment/Configurations и указать директорию синхронизации на сервере. Пример данной настройки приведён на рисунке 3.2.

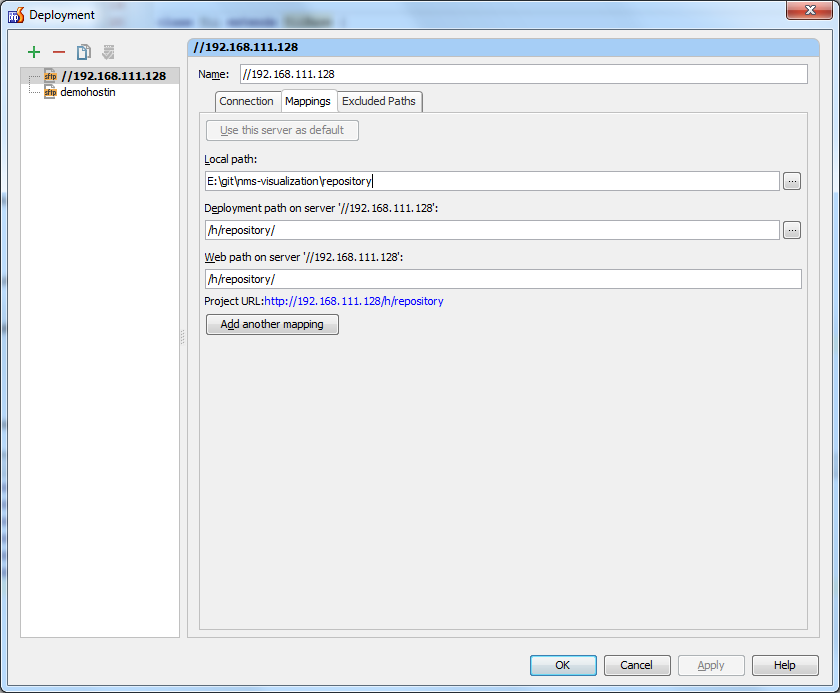


Рисунок 3.2 – Пример настройки синхронизации проекта

## Реализация модуля Ingraph

Установка Agavi Framework

Использую описанную выше настройку сервера, развернуть Agavi Framework достаточно просто. Для этого необходимо скачать пакет Agavi и с помощью PHP Storm загрузить его на сервер. Стандартный пакет Agavi имеет структуру, представленную на рисунке 3.3.

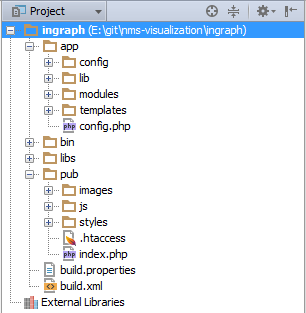


Рисунок 3.3 – Структура Agavi Framework

Директория app содержит исполняемые файлы, файлы конфигурации фреймворка, а так же файлы для формирования отображения страницы. Директория pub содержит элементы контроллера. В данной директории содержатся js-скрипты, изображения для сайта и CSS-файлы lzk настройки отображения. Директория libs содержит вспомогательные библиотеки. В нашем случаи таковой является Doctrine.

Для установки Doctrine необходимо скачать архив Doctrine и распаковать его в директорию libs. Далее необходимо сгенерировать модели базы данных для того, что бы работать с базой данных repository по средствам классов PHP. Для генерации моделей необходимо выполнить скрипт, листинг которого представлен на рисунке 3.4.

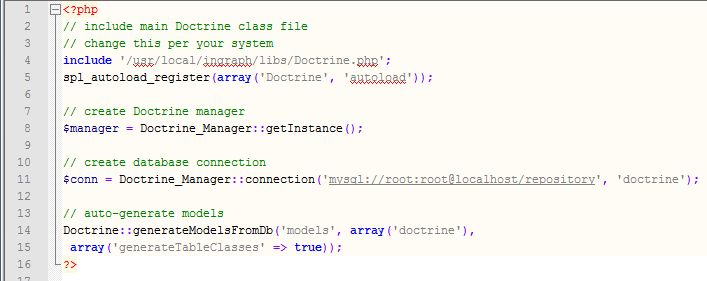


Рисунок 3.4 – Пример скрипта генерации моделей

После этого будут сгенерированы модели базы данных. Пример моделей приведён на рисунке 3.5.

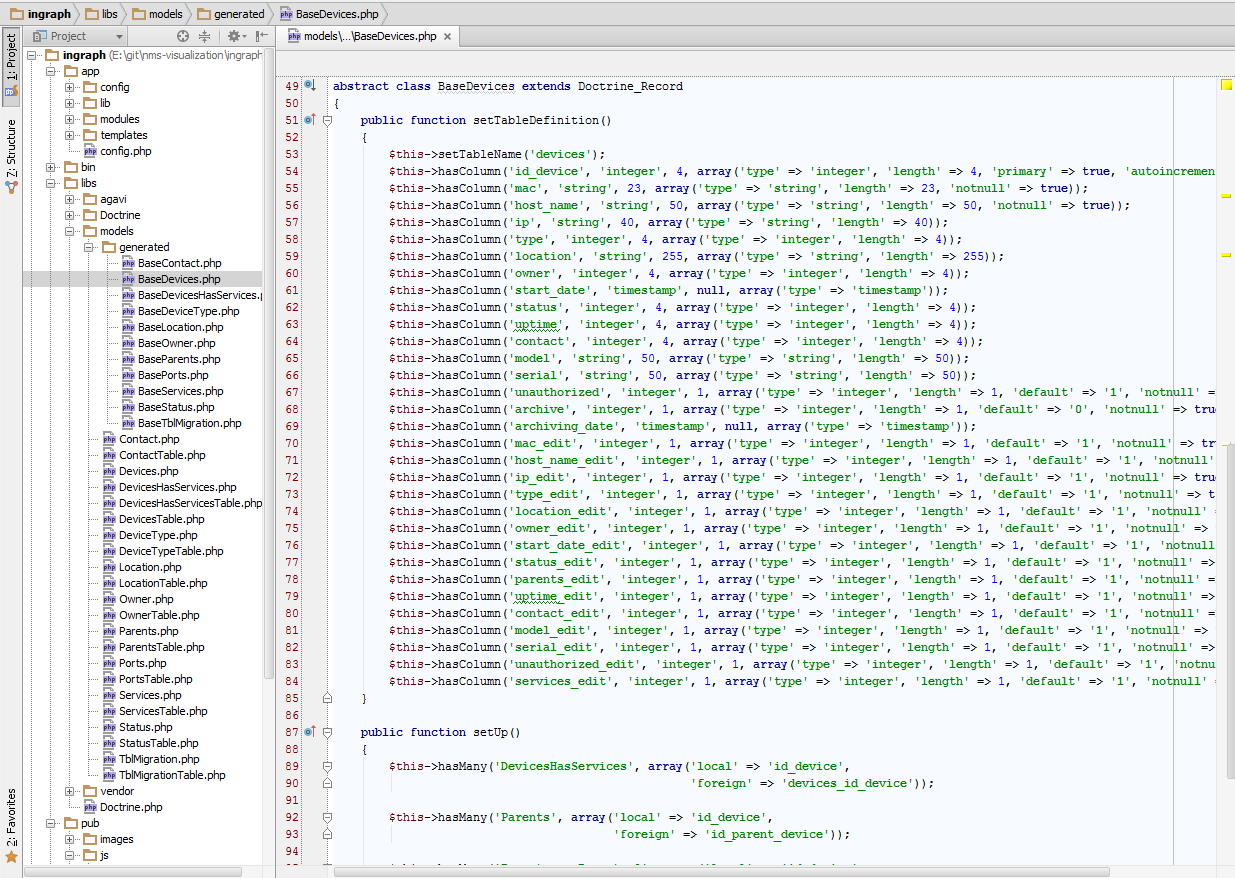


Рисунок 3.5 – Пример скрипта сгенерированной модели

Для подключение библиотеки Doctrine необходимо настроить скрипт конфигурации agavi database.xml (рисунок 3.6).

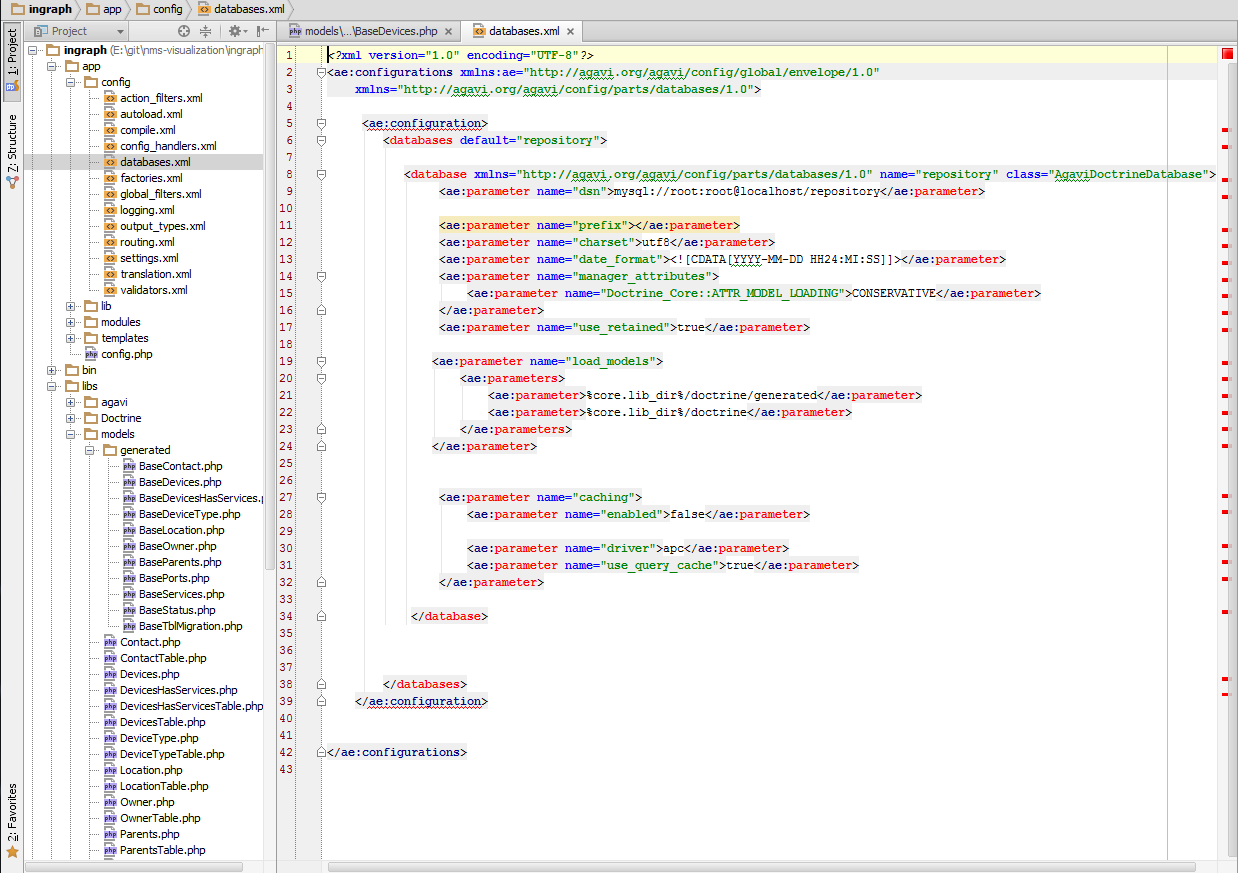


Рисунок 3.6 – Конфигурационный файл настройки подключения Doctrine

Выполнив эти действия, получим подключенную библиотеку Doctrine для работы с базой данных и модели классов, соответствующих таблицам базы данных.

Реализация web-приложения.

Для реализации поставленной задачи, прежде всего был установлен и настроен модуль Flot. Так же для данного модуля были реализованы следующие вспомогательные функциональные части:

* jquery.flot.description;
* jquery.flot.offset;
* jquery.flot.thresholds;
* jquery.flot.highlight.

Рассмотрим более подробно реализованные моули.

Jquery.flot.description. Данный модуль необходим для отображения вспомогательной информации на графике. Данный модуль работает с объектом Canvas и прорисовывает на нём буквы (именно прорисовывает, а не выводит текст). Пример данного вспомогательного модуля приведён на рисунке 3.7.

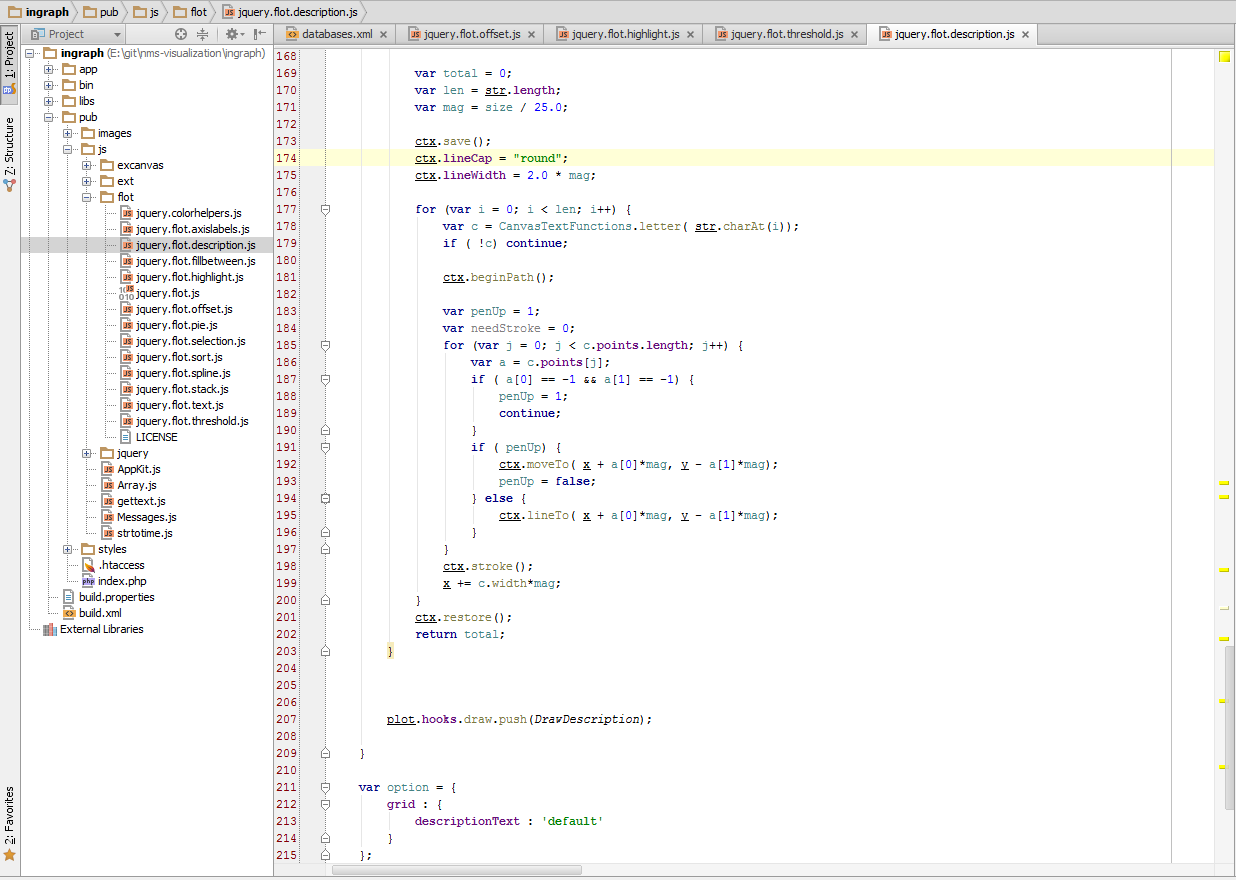


Рисунок 3.7 – Пример исходного кода вспомогательного модуля

Значения для вывода берутся из json-объекта, который присылает provider/values.

Jquery.flot.offset. Данный модуль предназначен для прорисовки сопряжённых графиков устройств. Данные о сопряжённом графике приходят в отдельном объекте. Прорисовка происходит путём смещения исходного положения canvas на 10% относительно текущей позиции.

Jquery.flot.thresholds. Данный модуль отображает пороговые значения для графиков.

Jquery.flot.highlight. Данный модуль прорисовывает подсветку точек. При наведение на точку, она выделяется и во всплывающем окне высвечивается значение по осям x и y.

Модуль подключаются путям добавления исполняемой функции в виде хука на определённое событие. Например:

plot.hooks.draw.push(ReturnOfset);

Так же был изменён модуль формирования данных json-объекта. Данный модуль получает данные из базы данных repository и ingraph. Пример скрипта для формирования данных представлен на рисунке 3.8

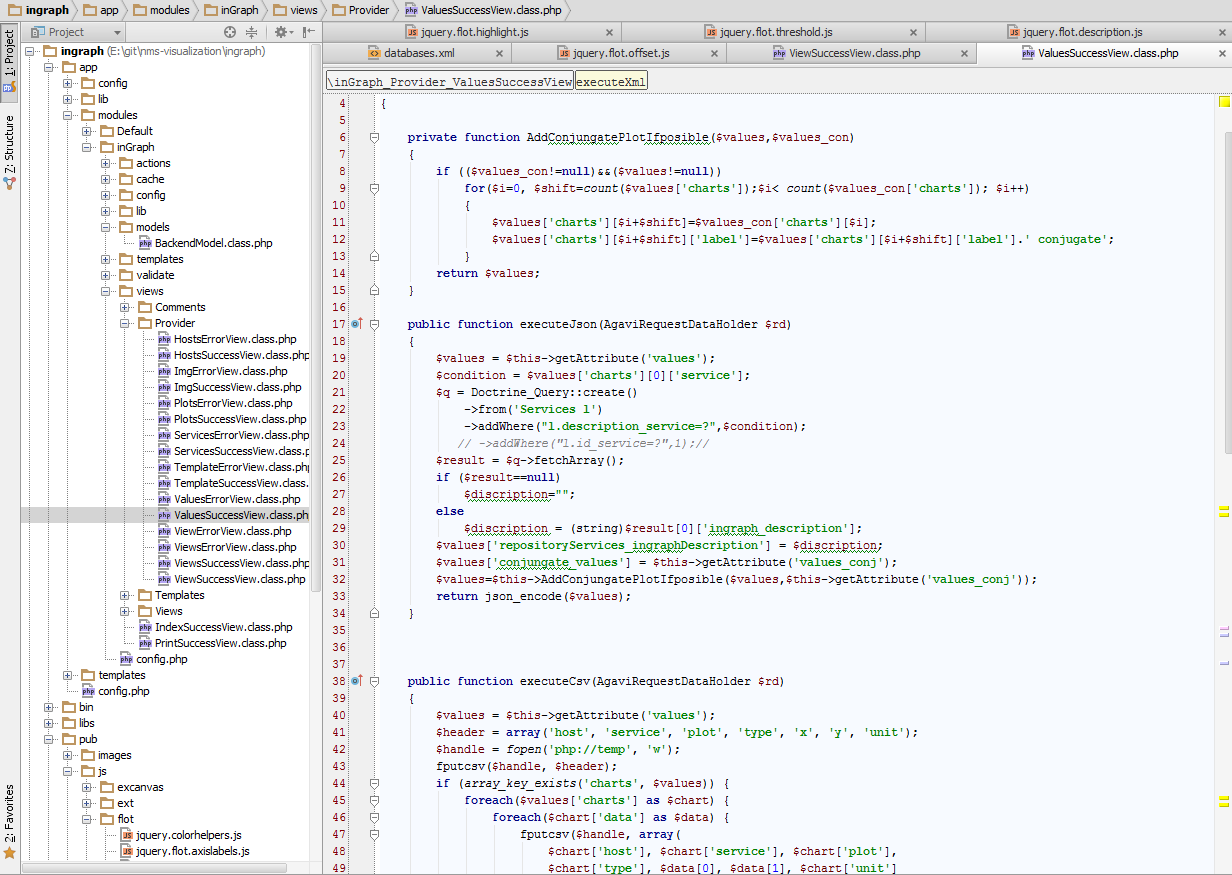


Рисунок 3.8 – Пример скрипта формирование ответа в виде json-объекта

Полный исходный код приложения представлен на диске в директории src.

## Реализация модуля Icinga

Данный модуль был реализован на основе готового приложения по сбору информации Icinga v1.10.2. Значительные работы были проведены в изменении графического интерфейса приложения. Модуль Icinga устанавливается из исходников проекта. Конфигурация проекта происходит следующей командой:

./configure --enable-MySQL --enable-ssl --enable-idoutils -- with-cgiurl=/usr/bin/Icinga

После этого производится установка следующей командой:

make all

make install

Структура работы Icinga имеет вид, представленный на рисунке 3.9.

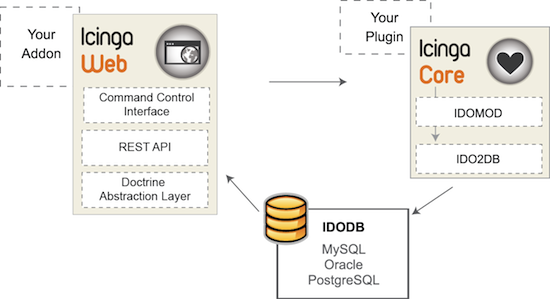


Рисунок 3.9 – Принцип работы Icinga

Для изменения графического интерфейса приложения необходимо реализовать приложения на С/С++, скомпилировать их и положить в директорию, в которой находятся cgi-приложения Icinga.

Пример реализации cgi-приложения приведён на рисунке 3.10.

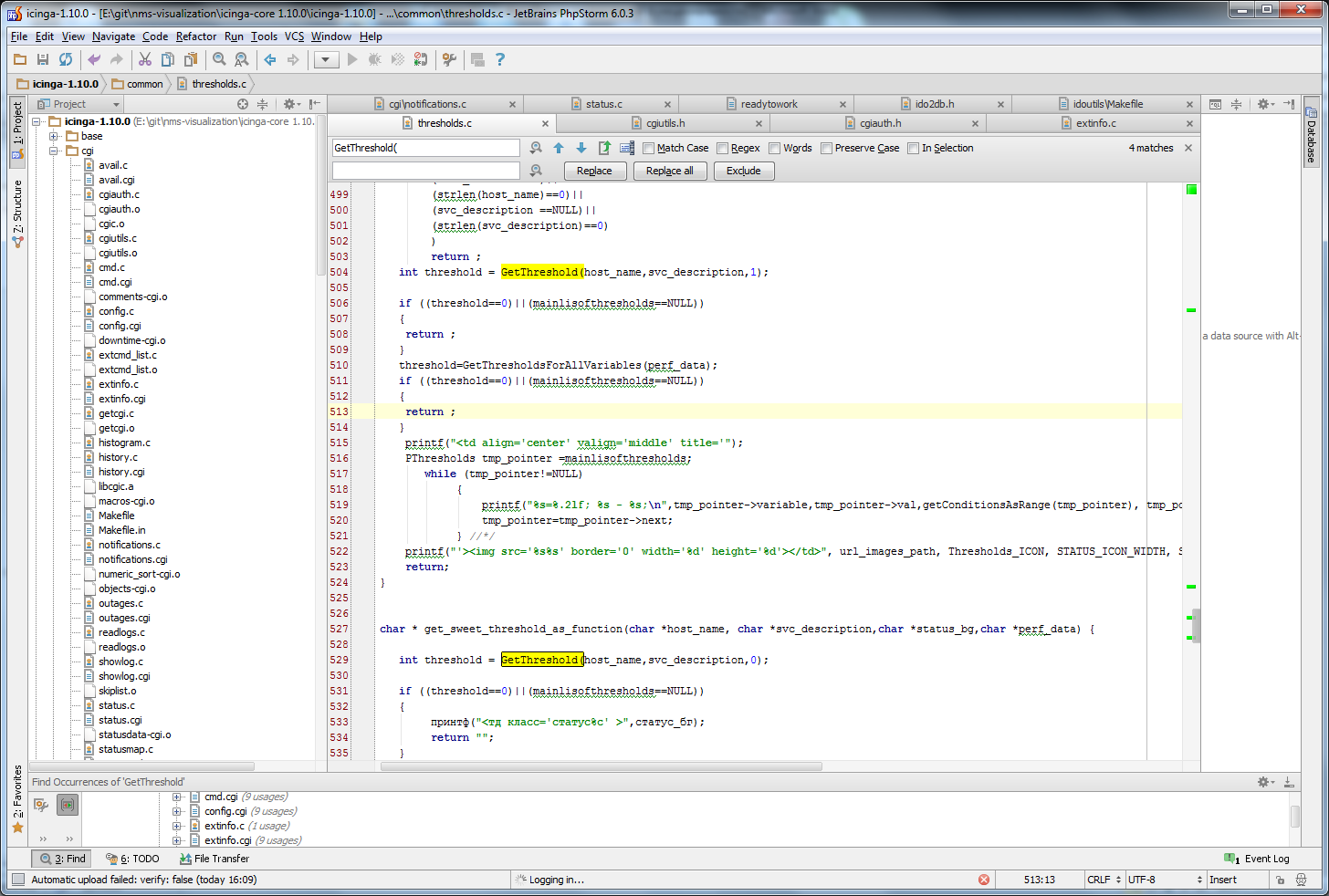


Рисунок 3.10 – Пример реализации приложения на С/С++

В ходе реализации приложения были решены следующие проблемы:

* добавлено отображение пороговых значений;
* добавлены фильтры к нотификация;
* добавлены отображения суммарных количеств записей и переход по страницам;
* добавлено отображение локаций для сервисов и хостов.

Так же ряд задач был решён с использование PHP скриптов, и ajax-запросов реализованных на JavaScript. К этим задачам можно отнести следующие:

* корректное отображение длинных строк;
* сохранение пользовательского интерфейса;
* динамическое изменение размеров колонок и их порядка;
* отображение мониторинга представляемых значений сервиса в реальном режиме времени;
* экспорт данных в pdf, png, docx, xlsx форматы.

Пример реализации скрипта представлен на рисунке 3.11.

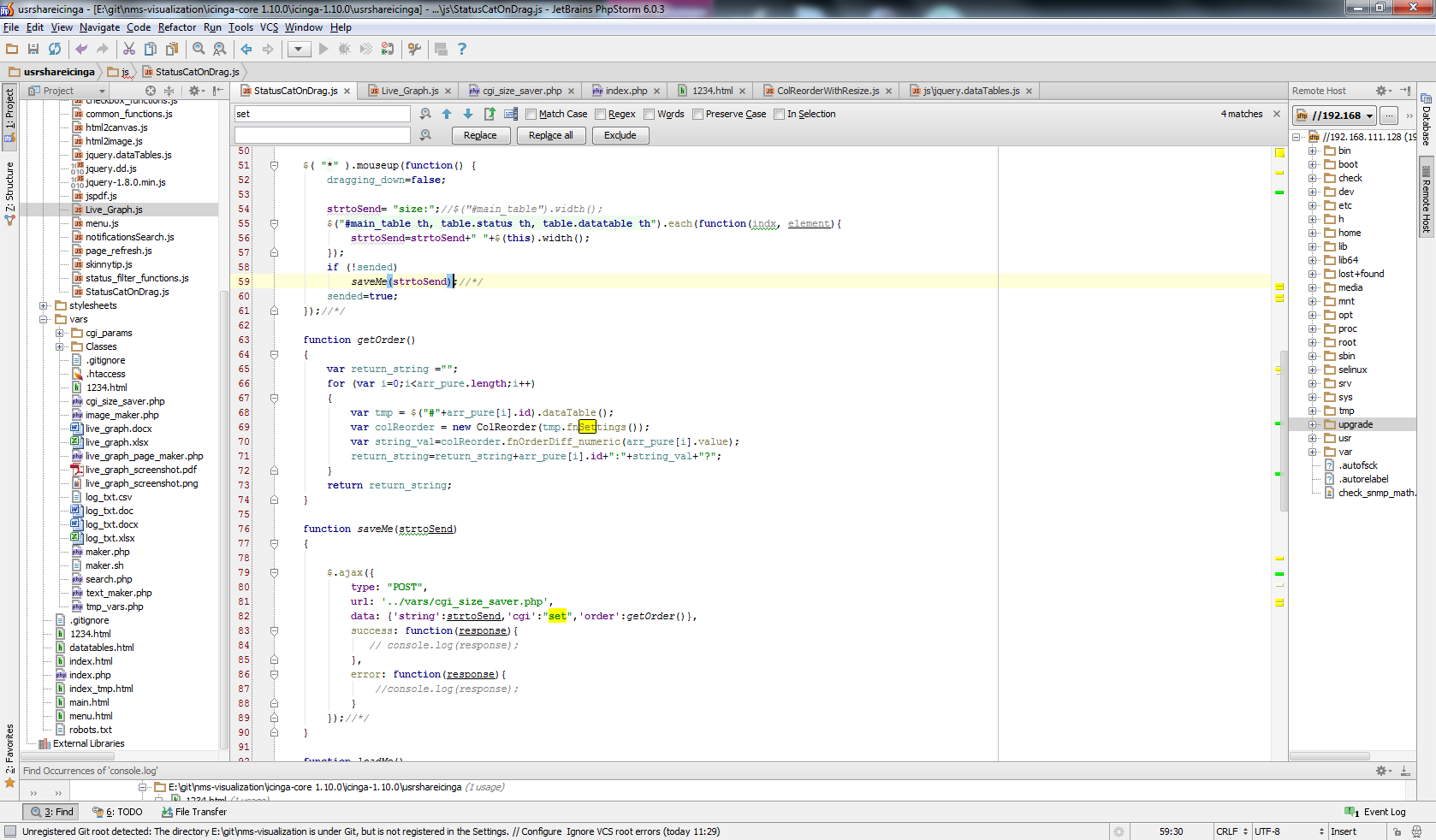


Рисунок 3.11 – Пример реализации скрипта на Javascript

Пример экспорта представлен в приложении Е.

Для удобной инсталляции обновлений был реализован скрипт readytowork, представленный на рисунке 3.12.

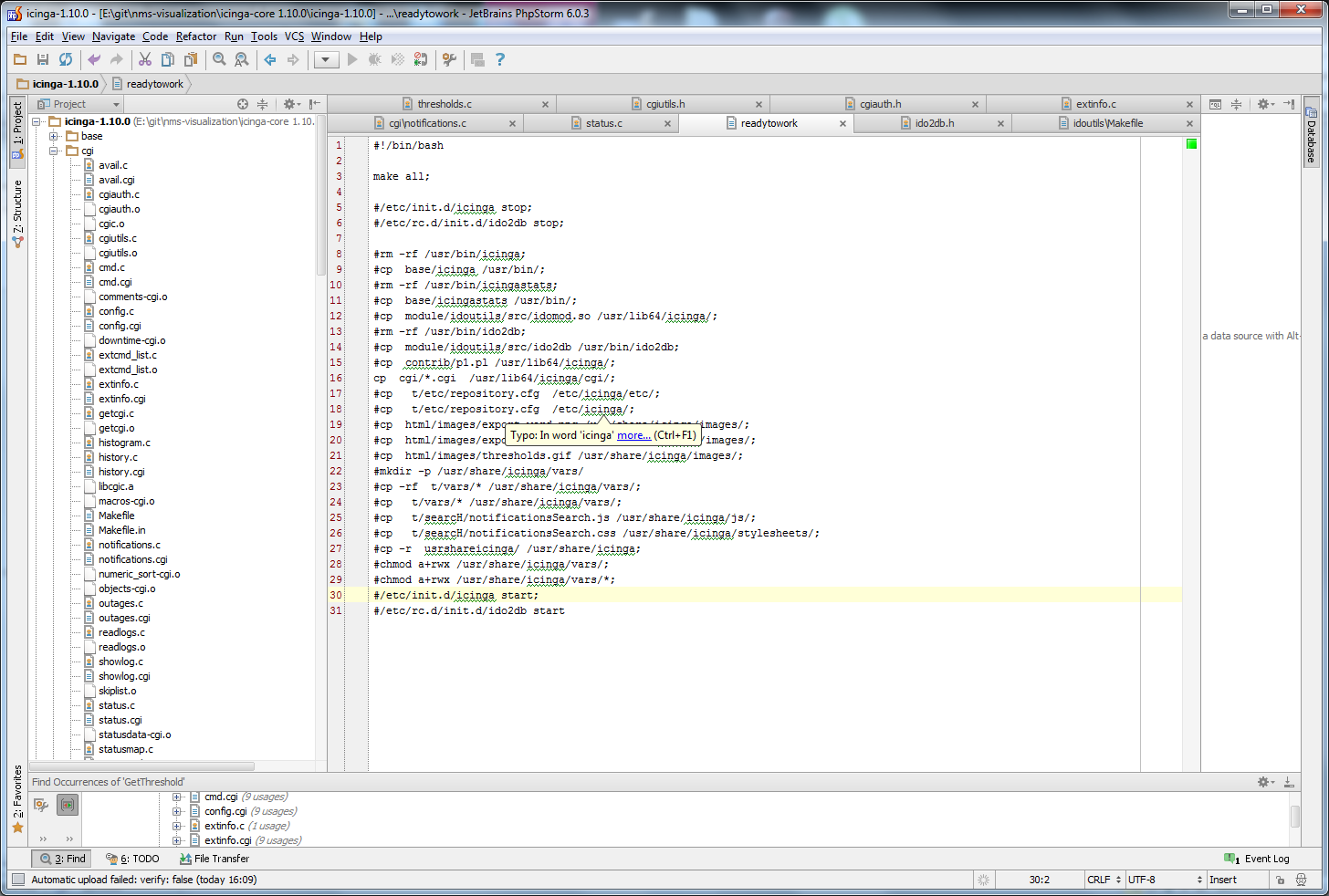


Рисунок 3.12 – Пример реализации скрипта на readytowork

## Реализация модуля AssetDB

Данный модуль необходим для настройки структуры локальной вычислительной сети а так же для внесения изменений в дополнительные параметры мониторинга ЛВС.

Как было описано выше данный модуль базируется на Yii-framework.

Для установки Yii достаточно скопировать исходники проекта в нужную директорию. Значительна часть проекта настраивается с использованием GUI фреймворка. К таким задачам относятся:

* создание моделей БД;
* создание CRUD (create - read - update - delete) для таблиц БД.

Пример GUI Yii-framework представлен на рисунке 3.13.

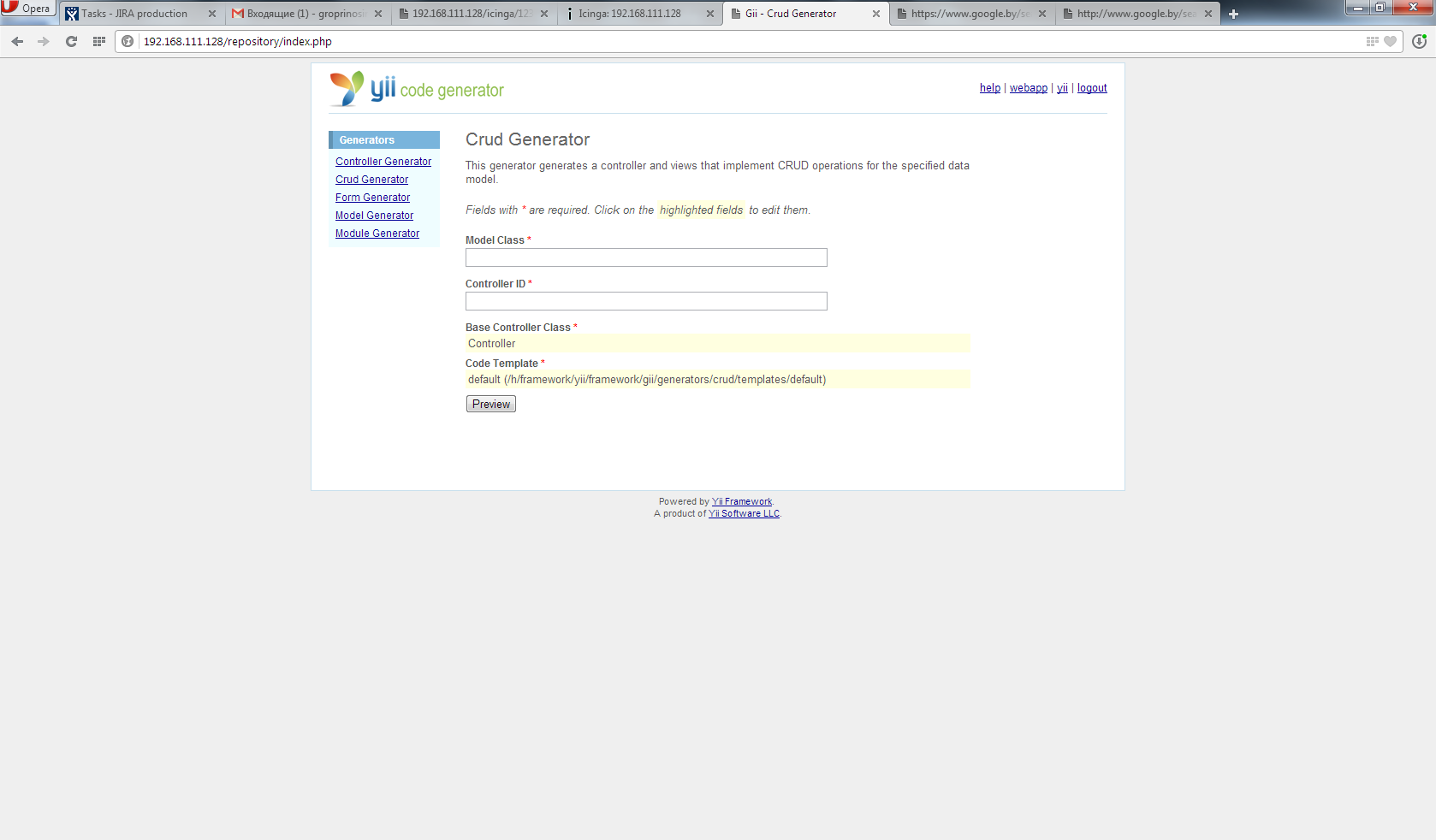


Рисунок 3.13 – Пример GUI Yii-framework

Для реализации интерфейса необходимо реализовать файлы view-модуля в директории /protected/view/<model>/<type>, где <model> – модель таблицы или связи БД и <type> – тип файла, определяющий доступ (полный admin, частичный index и т.д).

Пример реализации отображения представлен на рисунке 3.14

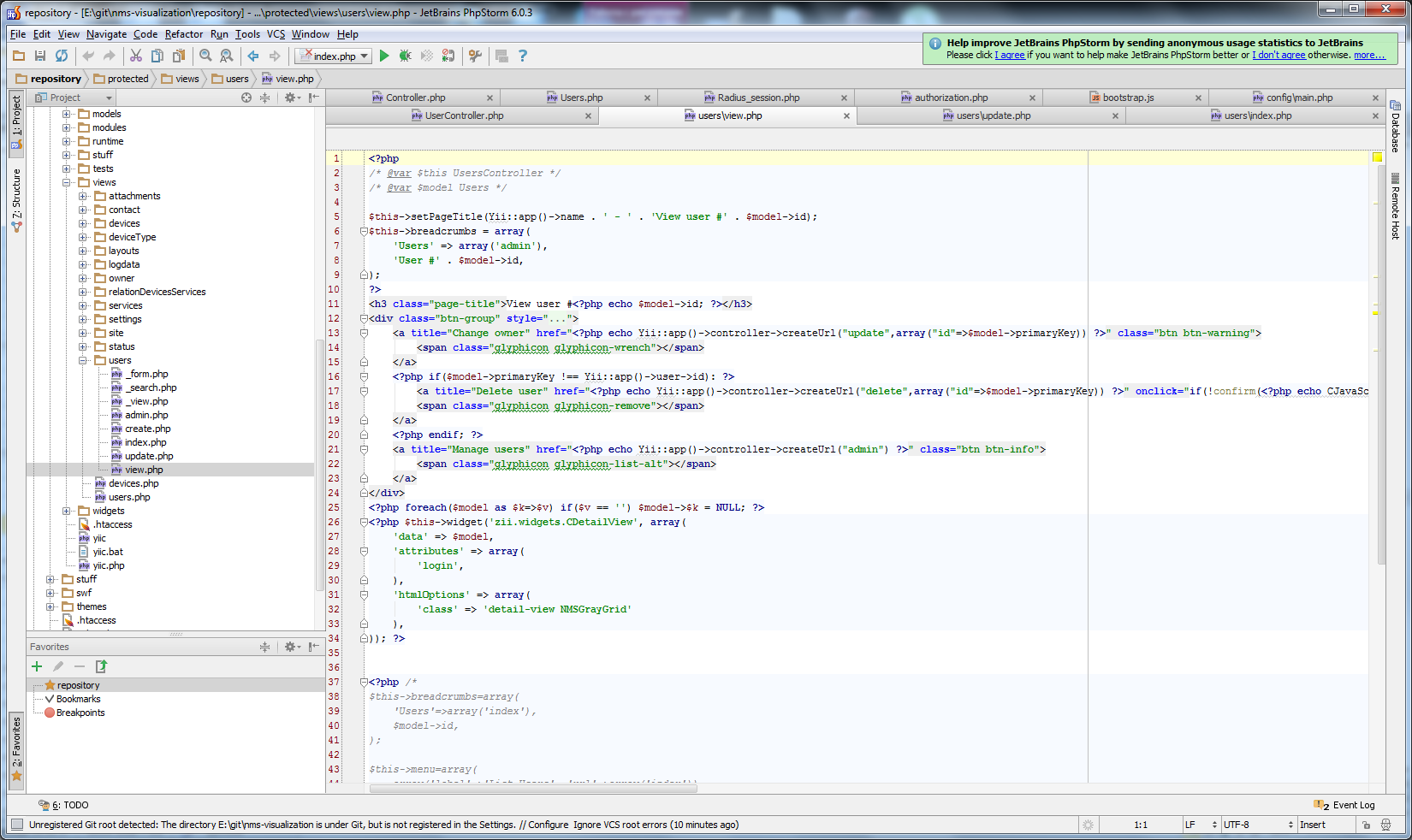


Рисунок 3.14 – Пример реализации представления

Так же необходимо реализовать контроллеры для обработки команд от пользователя. Это делается в директории /protected/controller/

Пример реализация контроллера приведена на рисунке 3.15.

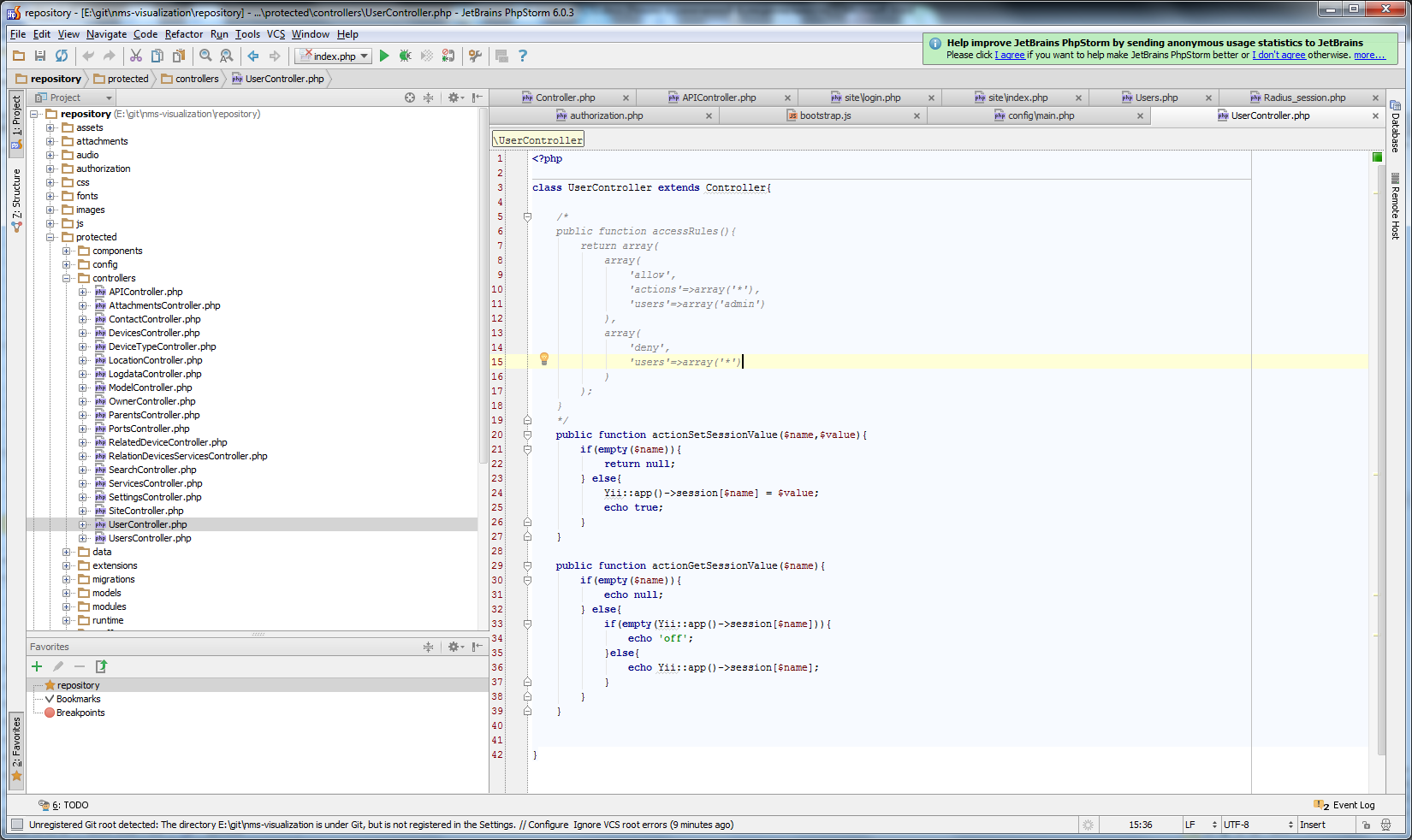


Рисунок 3.15 – Пример реализации контроллера

Реализованное приложение позволяет проводить полный контроль над базой данных repository, где хранятся все настройки мониторинга ЛВС.

## Тестирование программного продукта

Локальное тестирование проходило с использованием браузеров Opera и InternetExplorer по методу белого ящика. Это позволило устранить значительную часть неисправностей и неточностей приложения. Значительная часть тестируемой информация была получена в виде json-объектов. Пример отладочной информации приведён на рисунке 3.16.

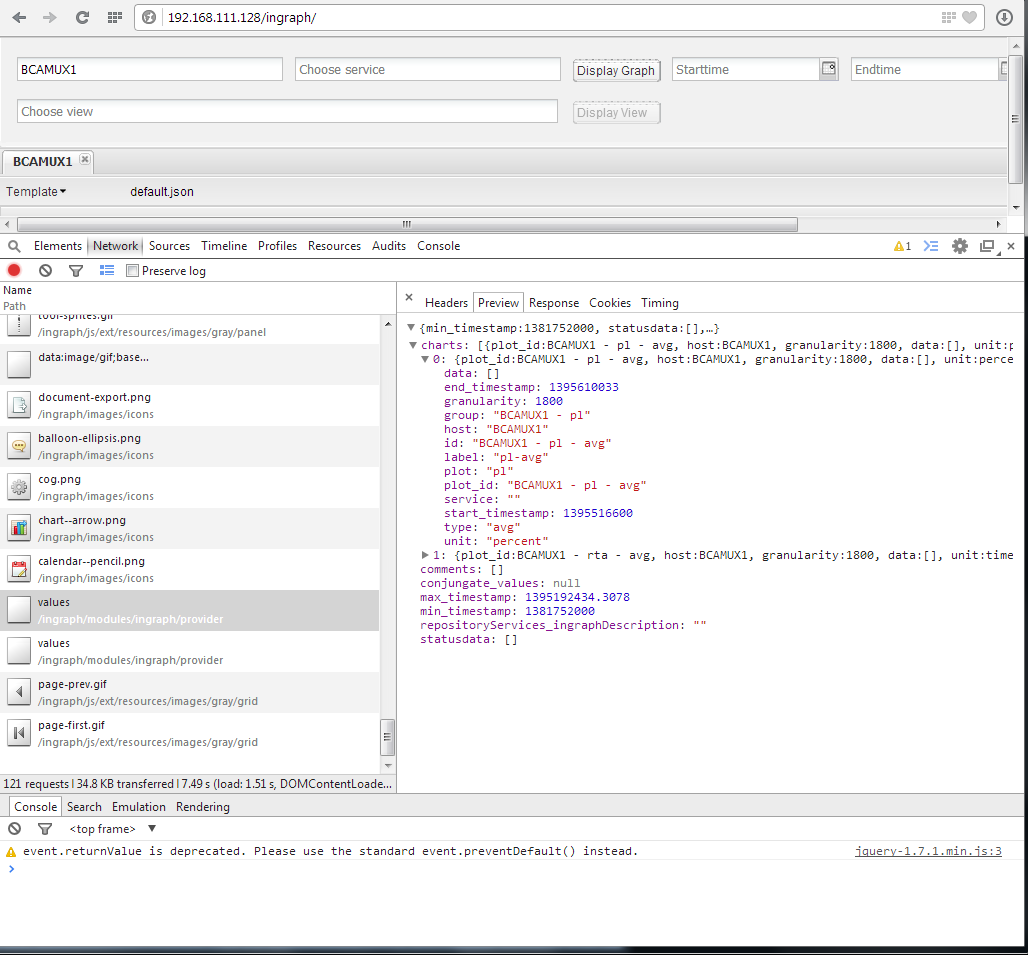


Рисунок 3.16 – пример отображения тестовой информации

Основное тестирование проходило по методу чёрного ящика профессиональными тестировщиками компании A1QA. В результате тестирования были найдены следующие недочёты для модуля Ingraph:

* [не выводятся графики при разном количестве выводимых параметров в parametrs data](https://jira.itransition.com/browse/NMS-257);
* [не отображается график](https://jira.itransition.com/browse/NMS-252) в Interne Explorer v 6;
* [отсутствует скроллинг влево и вправо при уменьшении окна браузера](https://jira.itransition.com/browse/NMS-251);
* [выводится не полное значение Ingraph description](https://jira.itransition.com/browse/NMS-250).

В результате тестирования модуля Icinga были выявлены следующие недочёты:

* изменение формата вывода пороговых значений;
* при выводе пороговых значений поле needOutput в БД repository учитывается не корректно;
* ошибка работы скрипта Live\_check в IE v6.

Данные неисправности были устранены и на данный момент рабочую версию приложения можно увидеть на тестовом сервере <https://nms1.cs.demohoster.com/ingraph>.

Пример работы готового приложения можно увидеть на рисунках 3.17 – 3.22.

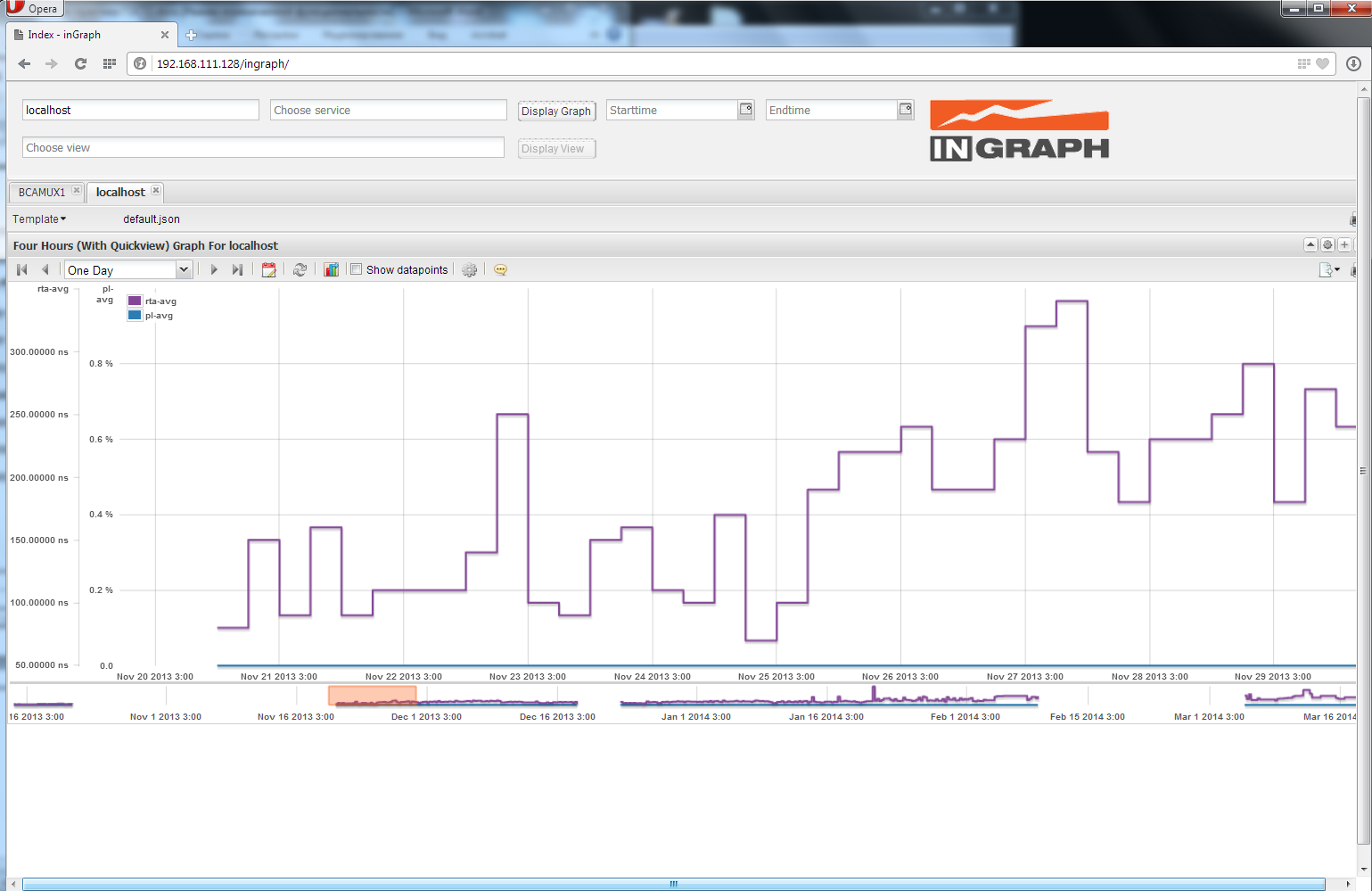


Рисунок 3.17 – Пример работы модуля Ingraph

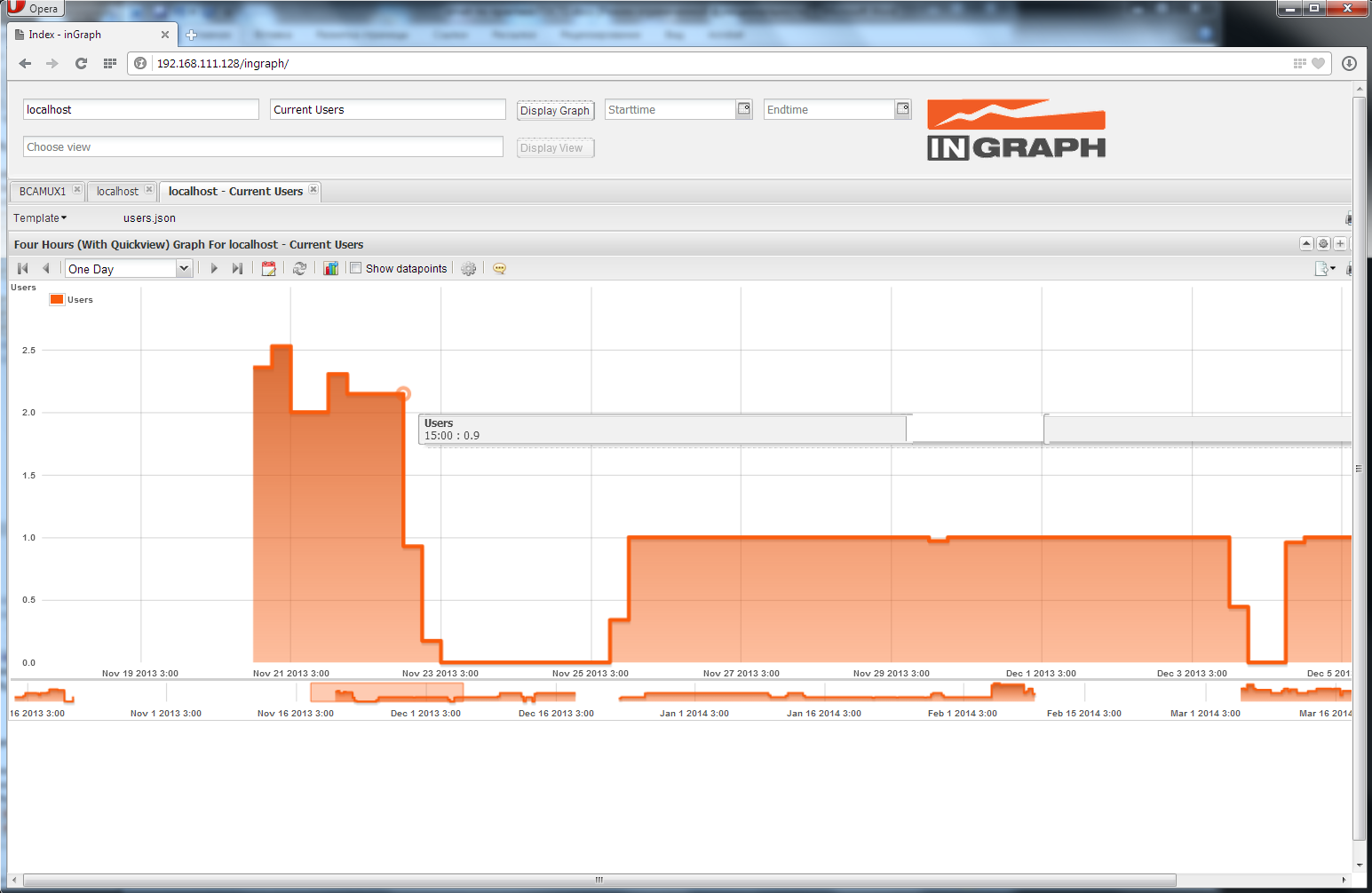


Рисунок 3.18 – Пример работы модуля Ingraph

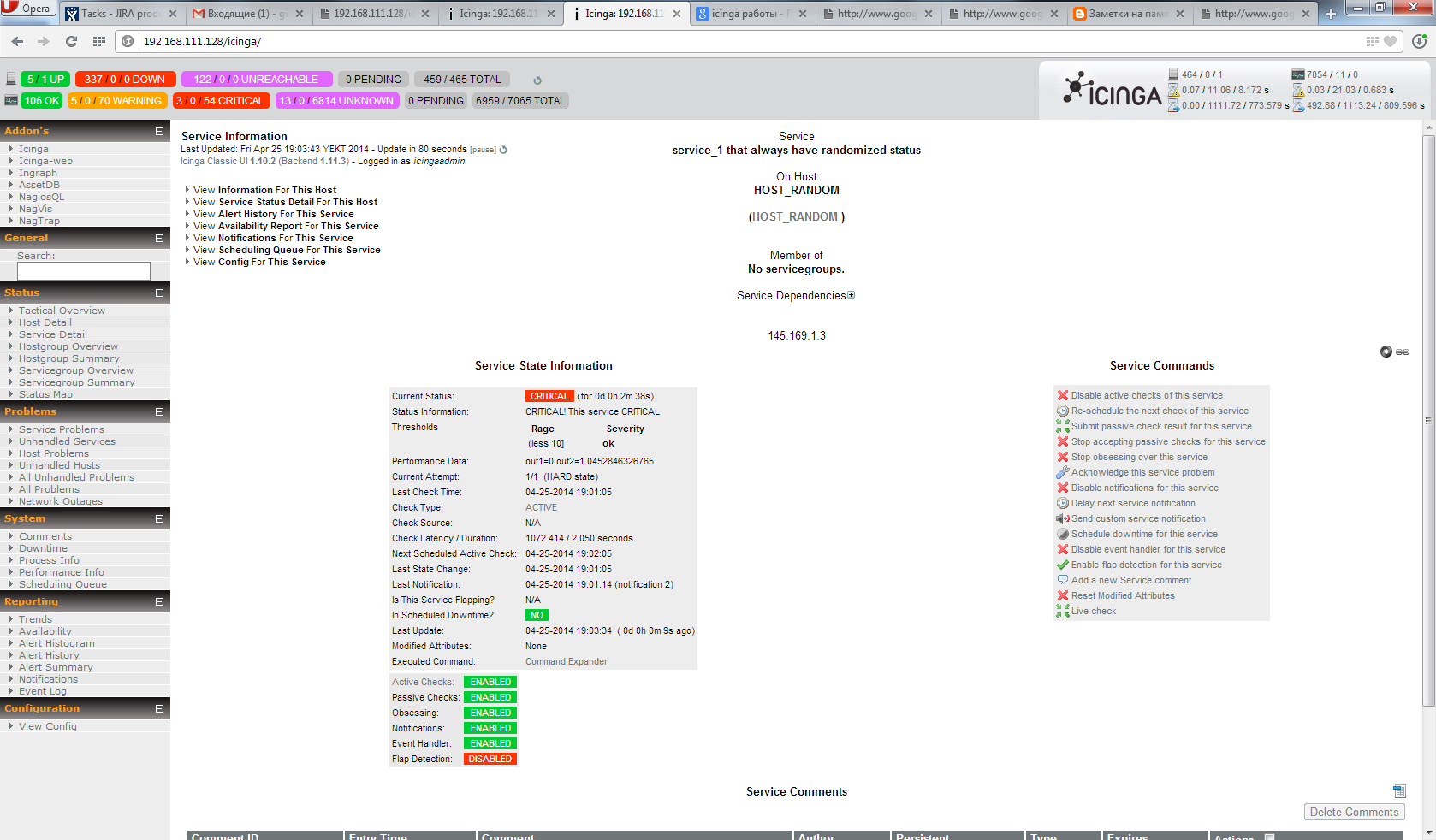


Рисунок 3.19 – Пример работы модуля Icinga

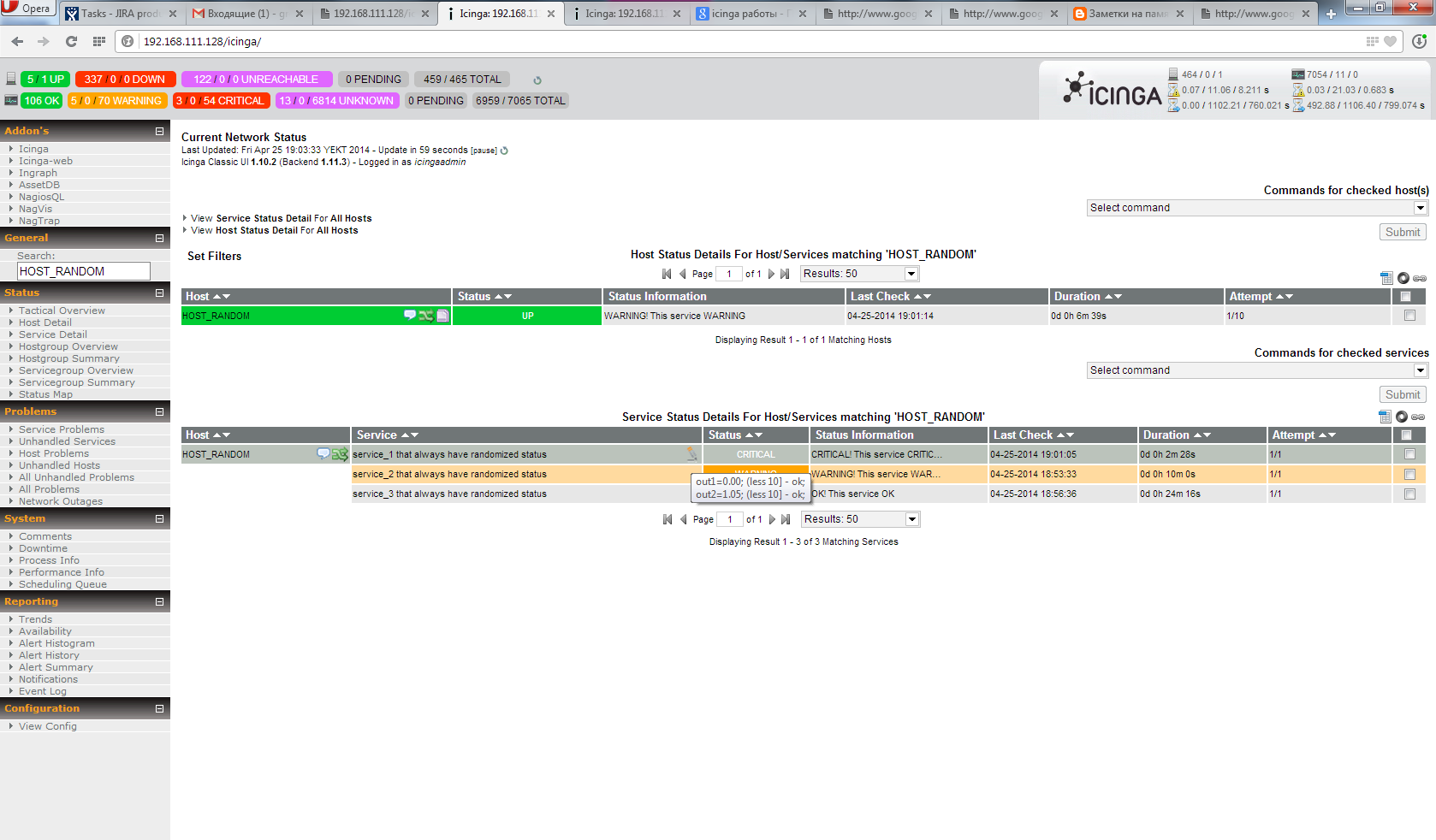


Рисунок 3.20 – Пример работы модуля Icinga

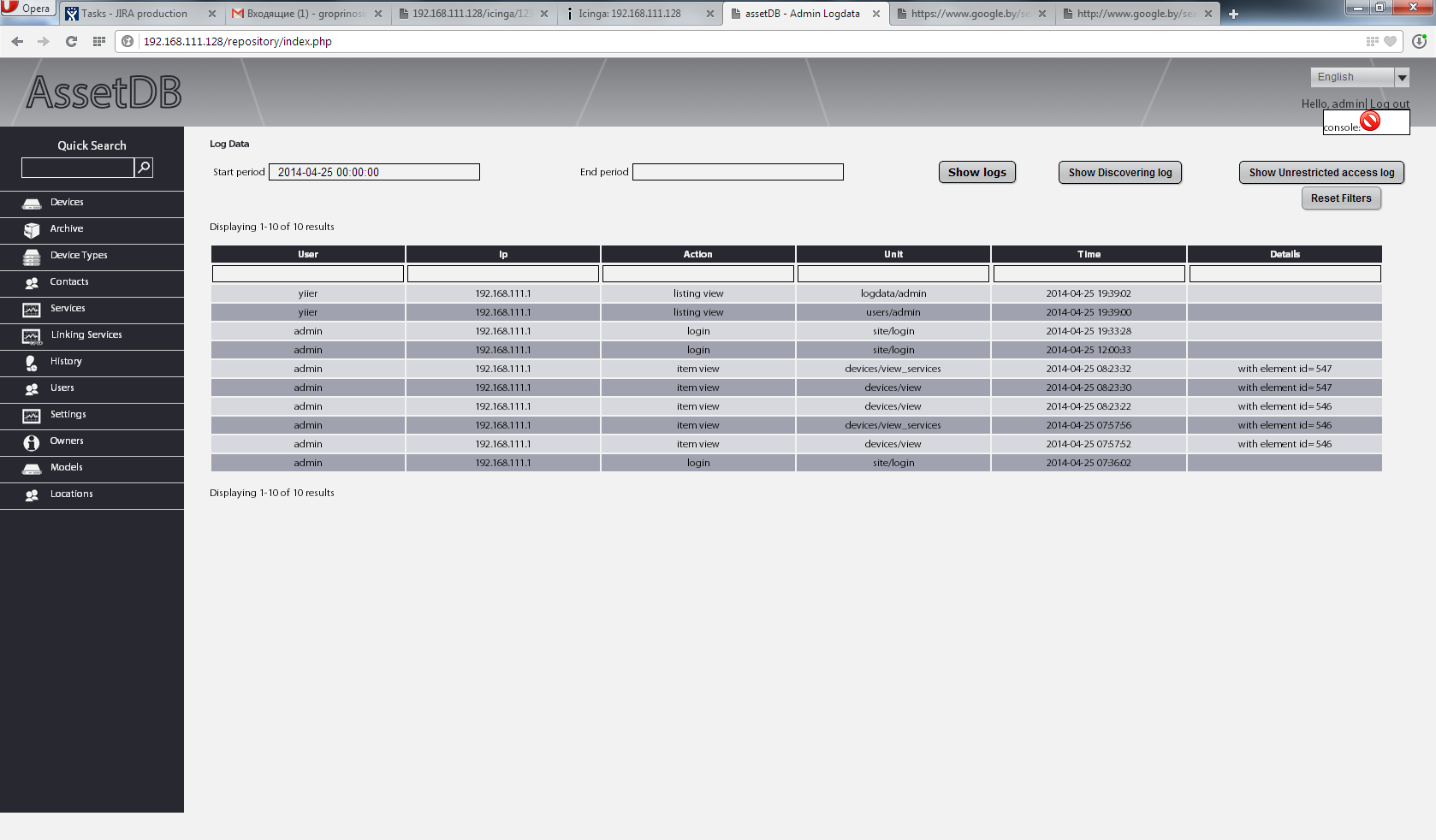


Рисунок 3.21 – Пример работы модуля AssetDB

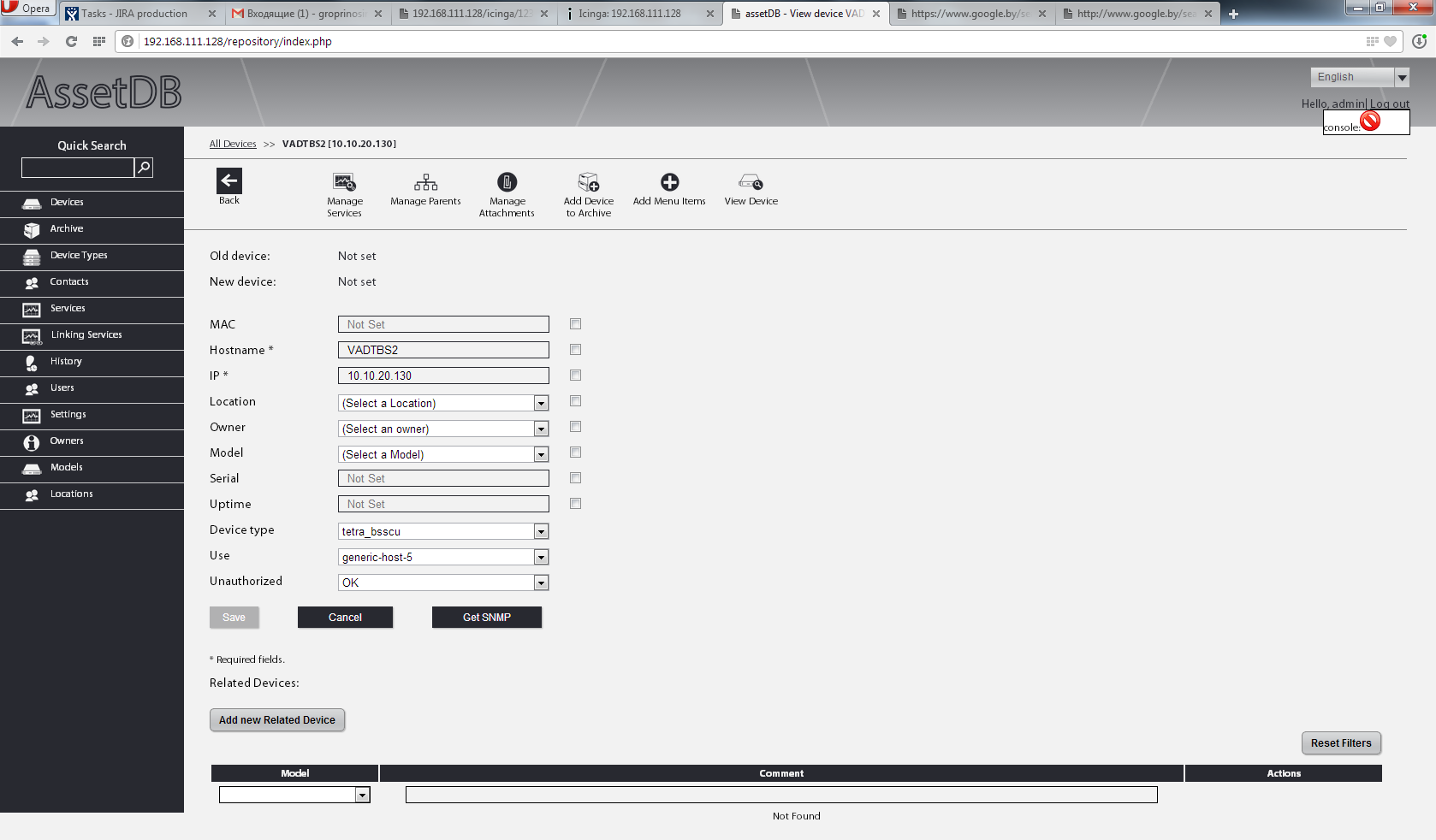


Рисунок 3.22 – Пример работы модуля AssetDB

# 4 Экономическая часть

## Обоснование необходимости создания системы мониторинга компьютерных систем и сетей

В рамках дипломного проекта разрабатывается система мониторинга устройств и сервисов нефтяного месторождения. Реализация данной системы позволит автоматизировать процесс проверки устройств, отвечающих за добычу и транспортировку нефти.

Корректно разработанная система позволяет оповещать операторов о нарушении работы комплекса в короткие сроки. Это позволяет существенно снизить риск глобальных сбоев и отказов системы. Вовремя полученное оповещение о выходе из строй устройства позволит принять меры по исправлению ситуации. Так же приложение даёт возможность сконфигурировать работу устройств и сервисов. Сбор данных ведётся круглосуточно и хранится на протяжении нескольких лет.

Разработанная система существенно снижает работу операторов по выявлению неисправностей.

Несмотря на широкий выбор систем мониторинга, достаточно сложно найти информативную, масштабную, удобную систему, включающую весь описанный функционал. Аналогом разрабатываемой системы является ZabbiX. Однако, описанный функционал приложения является платным. Разрабатываемая система базируется на модуле Icinga, которая является открытым проектом со свободно распространяемым исходным кодом. Это существенно снижает затраты на разработку и установку ПО на месторождении.

К основным задачам разрабатываемой системы, по мимо мониторинга, можно отнести конфигурацию устройств, иллюстрацию собранных значений, просмотр собранных параметров с датчиков в реальном режиме времени, экспорт данных в удобные форматы, конфигурация пользовательского интерфейса, регистрация действий пользователя в системе.

## Этапы работ по созданию системы мониторинга компьютерных систем и сетей

Разработка включает в себя все работы по созданию ПО и его компонент в соответствии с заданными требованиями. Включает оформление проектной и эксплуатационной документации, подготовку материалов, необходимых для проверки работоспособности и соответствующего качества программных продуктов, материалов, необходимых для организации обучения персонала и т.д.

Разработка ПО включает в себя следующие этапы:

* проектирование;
* реализация;
* тестирование и испытание системы;
* доработка;
* развёртывание/сопровождение.

К этапу «Проектирование» относится анализ технического задания на разработку и проектирование. На данный этап отводится значительный период времени, так как качественно выполненное проектирование позволяет сократить временные затраты на последующие этапы разработки.

К этапу «Реализация» относится непосредственно разработка программного продукта. Данный этап является самым продолжительным и составляет от 40 до 60% от всего времени разработки.

На этапе «Тестирование и испытание системы» проводится проверка соответствия результата этапа реализации результатам, полученным на этапе проектирования. По продолжительности данный этап занимает около 20% всего времени разработки.

На этапе «Доработка» проводится заключительные приемо-сдаточные испытания, при необходимости – внесение незначительных исправлений в программный продукт, и оформление заключительных глав пояснительной записки.

На этапе «Развёртывание/Сопровождение» производится поставка готового продукта заказчику, а так же установку готового продукта на ЭВМ заказчика, обучение операторов работе с новым ПО.

Графическое отображение этапов разработки программы представлено на рисунке 4.1.

Рисунок 4.1 – Диаграмма распределения времени работы при разработке проекта

Кроме деления жизненного цикла ПС ЭВТ на стадии и фазы, можно выделить восемь видов работ, которые выполняются в процессе создания программного продукта:

* анализ требований и планирование – предусматривает разработку спецификаций, анализ и модификацию функциональных, технических, интерфейсных требований;
* проектирование изделий – включает в себя определение, спецификацию, анализ и модификации аппаратно-программной архитектуры проекта программы и базы данных;
* программирование – детальное проектирование, кодирование, автономная отладка и комплексирование отдельных компонентов программы, а также планирование работ программистов, разработка базы данных, документирование отдельных компонентов и организация программирования на уровне компонентов;
* планирование отладки;
* верификация – процесс проверки корректности требований, отладки изделий и приёмные испытания;
* управление проектом – планирование и контроль проекта, контроль и регулирование договоров, связь с пользователями;
* контроль качества – разработка и контроль, стандартные и технические проверки программных средств и процессов разработки;
* документирование – разработка и корректировка руководства для пользователя и операторов.

Деление жизненного цикла программных средств на стадии, фазы и этапы, а также более подробное описание работ, выполняемых в процессе их разработки, производства и эксплуатации, необходимо для детального определения затрат, требуемых на эти цели.

## Расчет стоимости автоматизированной информационной системы

Для осуществления стоимостной оценки программного обеспечения и определения экономического эффекта необходимо составить смету затрат, включающую в себя следующие статьи расходов[4]:

* материалы и комплектующие ();
* электроэнергия ();
* основная заработная плата исполнителей ();
* дополнительная заработная плата исполнителей ();
* отчисления на социальные нужды ();
* амортизация ();
* расходы на спецоборудование ();
* накладные расходы ();
* прочие прямые расходы ().

Рассмотрим, что относят при расчете стоимости программного обеспечения на каждую из статей, перечисленных выше.

Статья «Материалы и комплектующие» отражает сумму затрат, связанных с расходом материалов на производство результатной информации (бумага, дискеты, красящие листы и т.д.).

Статья «Электроэнергия» включает затраты на электроэнергию.

К статье «Основная заработная плата исполнителей» относят затраты на заработную плату научных, инженерно-технических и других работников, непосредственно участвующих в разработке программных средств.

Статья «Дополнительная заработная плата исполнителей» включает оплату отпусков, времени выполнения государственных обязанностей, ученических дней и т.д. Дополнительная заработная плата принимается в размере 15% от основной заработной платы (по данным организации).

Статья «Отчисления на социальные нужды» включает в себя отчисления в фонд социальной защиты населения (Нсз – 34%) и отчисления на обязательное страхование от несчастных случаев (Нстр – 0,1-0,6%).

По статье «Амортизация» рассчитываются амортизационные отчисления, исходя из стоимости основных средств, используемых в процессе разработки программного обеспечения, сроков эксплуатации оборудования и годовой нормы амортизации.

Статья «Расходы на спецоборудование» включает расходы на специальное оборудование.

Статья «Накладные расходы» включает в себя затраты, связанные с необходимостью содержания аппарата управления, а также расходами на общехозяйственные нужды.

Статья «Прочие прямые расходы» на конкретное программное обеспечение включает затраты на приобретение и подготовку научно-технической информации и специальной литературы.

**Затраты по статье «Материалы и комплектующие»** учитывают расходы на разработку программного обеспечения (таблица 4.1).

Таблица 4.1 – Расходы на разработку программного обеспечения

|  |  |  |  |
| --- | --- | --- | --- |
| **Материал** | **Цена, руб.** | **Количество, шт.** | **Стоимость, руб.** |
| Печать листа формата А4 с бумагой | 500 | 200 | 100 000 |
| Бумага формата А1 | 10 000 | 6 | 60 000 |
| Компакт-диск  СD-RW | 4 800 | 2 | 9 600 |
| Всего: | 169 600 | | |

Таким образом, стоимость эксплуатационных материалов () составляет 169 600 рублей.

**Затраты на электроэнергию** определяются по формуле (4.1):

|  |  |
| --- | --- |
|  | (4.1) |

где  – стоимость 1 кВтч. Базовый тариф для юридических лиц с 1.03.2014 г. составляет 1 390,5 руб. за 1 кВтч (Согласно Декларации об уровне тарифов на электрическую энергию, отпускаемую республиканскими унитарными предприятиями электроэнергетики ГПО «Белэнерго» для юридических лиц и индивидуальных предпринимателей, зарегистрированной Департаментом ценовой политики Министерства экономики Республики Беларусь - приказ от 1.03.2014 г. №179)[14].

 – расход электроэнергии за один месяц;

 – период разработки программы, месяцев. Определяется в соответствии с длительностью дипломного проектирования и составляет 3 месяца.

Проведём расчёт требуемой мощности T. Прежде всего получим в среднем 8*·*21=168 рабочих часов в месяце. Один ПК в среднем потребляет 300 Вт. Для решения поставленной задачи необходимо 2 ПК и один сервер для тестирования. Потребление данного оборудования составит:

*3·0,3·168 = 151,2 кВтч.*

Так же учтём освещение. Современны энергосберегающие лампочки потребляют 20Вт. Для хорошего освещения помещения в 20 м2 достаточно 2х таких лампочек. Таким образом потребление на освещение составит:

*20·168·0,02·2 = 134,4 кВтч.*

Возьмём поправочный коэффициент 1,2 для определения затрат электроэнергии на другие нужды. Тогда получим:

*(151,2 +134,4) ·1,2=342,7 кВтч.*

Тогда затраты на электроэнергию составят:

*Э=342,7·3·*1 390,5 *= 1 429 500 руб.*

**Расчет по статье «Основная заработная плата»** (*Зо*) осуществляется по формуле:

|  |  |
| --- | --- |
|  | (4.2) |

где  – оклад специалиста, рублей. Специалисты, занимающиеся разработкой программного обеспечения, имеют 12 тарифный разряд. Поправочный коэффициент для 12 разряда равен 1,47. Тарифный коэффициент единой тарифной сетки для 12 разряда равен 2,84. На момент написания дипломного проекта (с  1 апреля 2014) тарифная ставка первого разряда равна 262 000 руб. Повышающий коэффициент равен 2.5. Таким образом, оклад специалиста составляет 2 734 500 рублей.

 – период разработки программы, месяцев. Определяется в соответствии с длительностью дипломного проектирования и составляет 3 месяца.

Премия составляет 10% от оклада.

Таким образом, основная заработная плата для одного исполнителя за 3 месяца составляет:

*Зо = 2 734 500·3+2 734 500·3·0,1=9 023 850 руб.*

**Дополнительная заработная плата** зависит от основной и вычисляется по формуле:

|  |  |
| --- | --- |
|  | (4.3) |

где  – дополнительная заработная плата;

 – основная заработная плата;

 – норматив дополнительной заработной платы на предприятии, выраженный в процентах (принимается равным 10%).

*Зд = 9 023 850·0,1 = 902 400 руб.*

**Затраты по статье «Отчисления на социальные нужды»** определяются в процентах от затрат на заработную плату. Сумма отчислений на социальные нужды рассчитывается по формуле:

|  |  |
| --- | --- |
|  | (4.4) |

где  – затраты на отчисления в Фонд социальной защиты населения и на обязательное страхование;

 – основная заработная плата;

 – дополнительная заработная плата;

– норматив отчислений в Фонд социальной защиты населения и отчислений на обязательное страхование, в процентах.

На данный момент согласно законодательству норматив отчислений в Фонд социальной защиты населения составляет 34%, а норматив отчислений на обязательное страхование от несчастных случаев примем в размере 1%. Следовательно, сумма отчислений составит:

*Зсз = (9 023 850+902 400)·0,35= 3 474 200 руб.*

**Прочие прямые расходы** - расходы на конкретное программное обеспечение включают затраты на приобретение и подготовку специальной научно-технической информации и специальной литературы. Определяются по нормативу, разрабатываемому в целом по организации, в процентах к основной заработной плате. К расходам данной группы можно отнести оплату интернет соединения и телефонной связи. При написании дипломного проекта используется безлимитный тариф стоимостью *156 000·3*=*468 000* бел. руб. за 3 месяца.

**По статье «Амортизация оборудования»** рассчитываются амортизационные отчисления () исходя из стоимости основных фондов, используемых в процессе разработки программного обеспечения, годовой нормы амортизации, сроков эксплуатации оборудования.

В процессе написания дипломной работы и разработки автоматизированной информационной системы мониторинга использовалось три ПК стоимостью 7 500 000 руб.

В таблице 4.2 приведен перечень со стоимостью программного обеспечения.

Таблица 4.2 – Стоимость программного обеспечения

|  |  |
| --- | --- |
| **Наименование** | **Стоимость, руб.** |
| Microsoft Office 2007 Профессиональный Russian | 4 265 600 |
| Microsoft Windows 7 Профессиональная Russian | 1 490 000 |
| Kaspersky Anti-Virus (на 2 ПК). Лицензия на 3 месяц | 73 250 |
| Всего | 5 828 850 |

Годовая норма амортизации равна 20% и для основных, и для нематериальных фондов. Действительное время работы оборудования по созданию программного обеспечения равно трем месяцам. Амортизационные отчисления для основных и нематериальных фондов рассчитаны в таблице 4.3.

Таблица 4.3 – Расходы на амортизацию

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Наименование** | **Первоначальная стоимость, руб.** | **Норма амортизации, %** | **Годовая сумма амортизации, руб.** | **Сумма амортизации на все время работы, руб.** |
| Персональный компьютер | 7 500 000*·*3= 22 500 000 | 20 | 22 500 000*·*0,2= 4 500 000 | 4 500 000/4= 1 125 000 |
| Программное обеспечение | 5 828 850 | 20 | 5 828 850*·*0,2= 1 165 750 | 1 165 750/4= 291 450 |
| Всего: | 1 416 450 | | | |

**Накладные расходы** рассчитываются по формуле:

|  |  |
| --- | --- |
|  | (4.5) |

где  – накладные расходы;

 – основная заработная плата исполнителей;

 – норматив накладных расходов.

Pн=9 023 850 \*10/100=902 400 руб.

На основании полученных данных по отдельным статьям затрат рассчитана плановая себестоимость программного обеспечения (таблица 4.4).

Таблица 4.4 – Расчет плановой себестоимости программного обеспечения

|  |  |
| --- | --- |
| **Статья затрат** | **Сумма затрат, руб.** |
| Материалы и комплектующие | *169 600* |
| Электроэнергия | *1 429 500* |
| Основная заработная плата | *9 023 850* |
| Дополнительная заработная плата | *902 400* |
| Отчисления на социальные нужды | *3 474 200* |
| Амортизация | *1 416 450* |
| Накладные расходы | *902 400* |
| Прочие прямые расходы | *468 000* |
| Общая сумма расходов по смете  (плановая себестоимость): | *17 786 400* |

Таблицу 4.4 удобно представить в виде диаграммы, отображающей влияние статей затрат на разработку программного обеспечения на окончательную себестоимость программы. Данная диаграмма представлена на рисунке 4.2.

Рисунок 4.2 – Структура затрат на разработку автоматизированной системы мониторинга

Таким образом, результатом данного раздела дипломного проекта является полный расчет количества денежных средств, затраченных на разработку программного обеспечения. Получаем, что полная (плановая) себестоимость равна сумме вышеперечисленных расходов, то есть равна *17 786 400* рублей.

## Расчёт цены разрабатываемой системы мониторинга компьютерных систем и сетей

Отпускная цена продукции формируется из плановой себестоимости, всех видов установленных налогов и прибыли, качества, потребительских свойств и конъюнктуры рынка. Отпускная цена продукции рассчитывается по формуле:

|  |  |
| --- | --- |
|  | (4.6) |

где  – отпускная цена;

 – плановая себестоимость;

 – прибыль.

Прибыль рассчитывается как произведение себестоимости и уровня рентабельности. Норма рентабельности составляет 0,3. Следовательно, прибыль составит:

*П=17 786 400·0,3=5 335 920 руб.*

Отпускная цена продукции составит:

*ОЦ = 17 786 400+ 5 335 100 = 23 121 500 руб.*

Стоимость проекта с учетом налога на добавленную стоимость представляет собой сумму отпускной цены и НДС. НДС рассчитывается по формуле:

|  |  |
| --- | --- |
|  | (4.7) |

где  – величина налога на добавленную стоимость;

 – ставка налога на добавленную стоимость, выраженная в процентах.

Ставка НДС с 2010 года установлена в размере 20%. Величина налога на добавленную стоимость составит:

*НДС = (23 121 500·20%)/100%=4 624 300 руб.*

Таким образом, цена программного средства с учетом НДС рассчитывается как сумма отпускной цены и величины налога на добавленную стоимость. Для разработанной автоматизированной системы цена с учетом НДС составляет:

*ОЦНДС= 23 121 500 +4 624 300 =27 745 800 руб.*

Величина *ОЦНДС*, равная *27 745 800* рублям, покрывает затраты и обеспечивает прибыль за разработку проекта.

## Сравнительный технико-экономический анализ эффективности производства нового продукта

Для сравнения проанализируем цены на схожие программные продукты. Стоит отметить, что полных аналогов данного продукта на рынке нет.

Наиболее ярким аналогом разрабатываемой системы является система мониторинга ZabbiX. Тот функционал, который требуется заказчику является платным в данной системе. Стоимость данного сертифицированного программного средства составляет 67 320 000 бел. рублей. В данном случае экономический эффект от внедрения разработки составит:

*67 320 000 - 27 745 800 = 39 574 200 руб.*

Таким образом, проанализировав рынок схожих программных продуктов, можно сделать вывод, что разработанная в ходе дипломного проектирования автоматизированная система более эффективна по техническим и экономическим показателям, чем рассмотренные выше программные продукты. Полученная расчетным путем стоимость разработанной автоматизированной системы находится на уровне низких показателей на рынке программных продуктов. По техническим параметрам разработанная автоматизированная система не уступает рассмотренным программным продуктам.

# Охрана труда

## 5.1 Введение

Разрабатываемая система мониторинга активности устройств в локальной сети предназначено для операторов ЭВМ, работающих на нефтяном месторождении. Таким образом при предоставлении документации о разрабатываемом проекте, необходимо рассмотреть вопросы охраны труда при работе с ПЭВМ, пожарную безопасность, электробезопасность а так же общие принципы организации труда на предприятии.

Охрана труда – система обеспечения безопасности жизни и здоровья работников в процессе трудовой деятельности, включающая правовые, социально-экономические, организационные, технические, психофизиологические, санитарно-гигиенические, лечебно-профилактические, реабилитационные и иные мероприятие и средства [13].

Важнейший социальный эффект от реализации мер по охране труда – это сохранение жизни и здоровья работающих, сокращение несчастных случаев и заболеваний на производстве.

Требования по охране труда – нормативные предписания, направленные на сохранение жизни, здоровья и работоспособности работников в процессе их трудовой деятельности, содержащиеся в нормативных правовых актах, в том числе технических нормативных правовых актах.

Закон Республики Беларусь “Об охране труда” был принят 23.06.2008 г. (Национальный реестр правовых актов РБ, 2008 г. №158,2/1453). Закон направлен на регулирование общественных отношений в области охраны труда и реализацию установленного Конституцией Республики Беларусь права граждан на здоровые и безопасные условия труда.

Статья 1 Закона содержит основные термины и их определения, применяемые в на стоящем Законе, к которым также можно отнести производственный фактор, вредный производственный фактор и опасный производственный фактор.

Согласно вышеупомянутому Закону:

**–** производственные факторы – факторы производственный среды и трудового процесса;

**–** вредный производственный фактор – производственный фактор, воздействие которого на работающего в определенных условиях может привести к снижению работоспособности и (или) заболеванию (в зависимости от уровня и продолжительности воздействия вредный производственный фактор может стать опасным);

**–** опасный производственный фактор – производственный фактор, воздействие которого на работающего в определенных условиях способно привести к травме или другому внезапному резкому ухудшению здоровья либо к смерти.

Опасные и вредные производственные факторы классифицируются по различным признакам. Согласно государственной классификации опасные и вредные факторы подразделяются на следующие группы:

**–** физические;

**–** химические;

**–** биологические;

**–** психофизиологические [1].

В соответствии с пунктом 53 Правил по характеру и времени проведения инструктаж по охране труда подразделяют на:

**–** вводный;

**–** первичный на рабочем месте;

**–** повторный;

**–** внеплановый;

**–** целевой.

## 5.2 Промышленная санитария и гигиена труда

При работе на ПЭВМ и другой офисной технике работающие при определенных условиях могут подвергаться воздействию различных опасных и вредных производственных факторов, основными из которых являются [2]:

**–** физические:

**–** повышенное значение напряжения в электрической цепи, замыкание

**–** которой может произойти через тело работающего;

**–** повышенные уровни электромагнитного излучения промышленной частоты и высокочастотные;

**–** повышенные уровни рентгеновского излучения;

**–** повышенные уровни ультрафиолетового излучения;

**–** повышенные уровни инфракрасного излучения;

**–** повышенные уровни статического электричества;

**–** повышенные уровни запыленности воздуха рабочей зоны;

**–** повышенное содержание положительных аэроионов в воздухе рабочей зоны;

**–** пониженное содержание отрицательных аэроионов в воздухе рабочей зоны;

**–** повышенная или пониженная температура воздуха рабочей зоны;

**–** повышенная или пониженная влажность воздуха рабочей зоны;

**–** повышенная или пониженная подвижность воздуха рабочей зоны;

**–** повышенный или понижения уровень освещенности рабочей зоны;

**–** повышенный уровень прямой и отраженной блесткости;

**–** неравномерность распределения яркости в поле зрения;

**–** повышенная или пониженная яркость светового изображения;

**–** повышенный уровень пульсаций светового потока;

**–** химические (повышенное содержание в воздухе рабочей зоны окиси углерода, озона, аммиака, фенола, формальдегида и полихлорированных фенилов);

**–** биологические (повышенное содержание в воздухе рабочей зоны микроорганизмов);

**–** психофизиологические:

**–** напряжение зрения;

**–** напряжение памяти;

**–** напряжение внимания;

**–** длительное статическое напряжение;

**–** большой объем информации, обрабатываемой в единицу времени;

**–** монотонность труда;

**–** нерациональная организация рабочего места;

**–** эмоциональные перегрузки.

Размеры рабочего помещения должны соответствовать количеству работающих и разрешаемому в них комплекту технических средств. В них предусматриваются соответствующие параметры температуры, освещения, чистоты воздуха, обеспечивают изоляцию, от производственных шумов. Для обеспечения нормальных условий труда санитарные нормы устанавливают на одного работающего, объем производственного помещения не менее 15 м3, площадь помещения выгороженного стенами или глухими перегородками не менее 4*;*5 м3 [2].

Гигиенические требования к микроклимату производственных помещений нормируются СанПиН 9-80 РБ 98. Оптимальные нормы микроклимата помещения приведены в таблице 5.1[11].

Таблица 5.1 – Оптимальные нормы микроклимата помещений

|  |  |  |  |
| --- | --- | --- | --- |
| **Период года** | **Температура воздуха, °С** | **Относительная влажность, %** | **Скорость движения воздуха, м/с** |
| холодный | от 21 до 23 включ. | от 40 до 60 включ. | 0,1, не более |
| теплый | от 22 до 24 включ. | от 40 до 60 включ. | 0,2, не более |

Поддерживание параметров микроклимата в помещении обеспечивается отоплением и кондиционированием. Климатические условия, поддерживаются в пределах:

**–**температура, от 15*◦* C до 30*◦* C;

**–**относительная влажность воздуха от 20 % до 80 %;

**–**концентрация пыли в воздухе не более 0*;*5 мг/м3.

В помещении ежедневно должна проводиться влажная уборка.

Рациональное цветовое оформление помещения направленно на улучшение санитарно-гигиенических условий труда, повышение его производительности и безопасности. Окраска помещений, в которых находится ПЭВМ, влияет на нервную систему человека, его настроение и в конечном счете на производительность труда. Освещение помещения и оборудования должно быть мягким, без блеска. Рабочие комнаты и кабинеты должны иметь естественное освещение.

Снижение шума, создаваемого на рабочих местах внутренними источниками, а также шума проникающего извне, является очень важной задачей. Под настольные шумящие аппараты можно подкладывать мягкие коврики из синтетических материалов, а под ножки столов, на которых они установлены, – прокладки из мягкой резины, войлока, толщиной 6–8 мм. Снижение уровня шума, проникающего в производственное помещение извне, может быть достигнуто увеличением звукоизоляции ограждающих конструкций, уплотнением по периметру притворов окон, дверей.

При работе монитора на экране кинескопа накапливается электростатический заряд, создающий электростатическое поле. При этом люди, работающие с монитором, приобретают электростатический потенциал.

По обобщенным данным, у работающих за монитором от 2 до 6 часов в сутки функциональные нарушения центральной нервной системы происходят в среднем в 4*;*6 раза чаще, чем в контрольных группах, болезни сердечно-сосудистой системы – в 2 раза чаще, болезни верхних дыхательных путей – в 1*;*9 раза чаще, болезни опорно-двигательного аппарата – в 3*;*1 раза чаще. С увеличением продолжительности работы на компьютере соотношения здоровых и больных среди пользователей резко возрастает. Даже при кратковременной работе длительностью 45 минут в организме пользователя под влиянием электростатического поля происходят значительные изменения гормонального состояния и специфические изменения биотоков мозга.

Допустимые уровни напряженности электростатических полей устанавливаются в зависимости от времени пребывания персонала на рабочих местах. Предельно допустимый уровень напряженности электростатических полей устанавливается равным 60 кВ/м в течение 1 ч. При напряженности электростатических полей менее 20 кВ/м время пребывания в электростатических полях не регламентируется. Контроль напряженности электростатических полей проводится в следующих случаях:

**–** при организации нового рабочего места;

**–**в порядке текущего надзора за действующими электроустановками и технологическими процессами;

**–**при вводе нового технологического процесса, сопровождающегося электризацией материалов.

Основные мероприятия, применяемые для защиты от статического электричества производственного происхождения, включают методы, исключающие или уменьшающие интенсивность генерации зарядов, и методы, устраняющие образующиеся заряды.

В основном из средств защиты предлагаются защитные фильтры для экранов мониторов. Они улучшает эргономические параметры экрана монитора и снижает излучение монитора в направлении пользователя.

Одним из путей улучшения условий труда можно рассматривать искусственную ионизацию воздуха, насыщение его легкими отрицательными ионами. Для этой цели применяются генераторы отрицательных ионов воздуха, иногда называемые аэроионизаторами, ионизаторами воздуха, люстрами Чижевского или лампой Чижевского. Применение генератора отрицательных ионов воздуха на рабочем месте оператора ПЭВМ позволяет смещать соотношение между положительными и отрицательными ионами в сторону отрицательных ионов, что положительно влияет на работоспособность.

Образующиеся заряды статического электричества устраняют путем заземления электропроводных частей оборудования. Сопротивление такого заземления должно быть не более 100 Ом. При невозможности устройства заземления практикуется повышение относительной влажности воздуха в помещении. Можно увеличить объемную проводимость диэлектрика, для чего в него вносят графит, ацетиленовую сажу, алюминиевую пудру, а в жидкие диэлектрики – специальные добавки.

К средствам индивидуальной защиты от статического электричества относятся электростатические халаты и специальная обувь, подошва которой выполнена из кожи либо электропроводной резины, а также антистатические браслеты.

## 5.3 Техническая безопасность

К опасным производственным факторам, влияющим на человека при работе с ПЭВМ можно отнести следующие:

**–** пожарная опасность, обусловленная наличием на рабочем месте мощного источника энергии. При коротких замыканиях по проводникам протекают большие токи, вследствие чего происходит нагрев проводников, загорается изоляция и окружающие предметы. Для снижения вероятности реализации этой опасности необходимо проверять изоляцию токоведущих проводов, соединений аппаратов, их исправность;

**–** высокое напряжение питающей сети, обуславливающее возможность поражения человека электрическим током. При работе в компьютерном классе оператор подвержен воздействию высокого напряжения. Источниками высокого напряжения в помещении являются ЭВМ, светильники, различная бытовая техника: множительная и копировальная, кондиционеры. Так же оператор может случай но прикоснуться к токоведущим частям. Опасность реализуется при непосредственном контакте с электрооборудованием. Оператор, подвергнувшийся кратковременному воздействию высокого напряжения, может испытать болевой шок, потерять сознание, получить ожоги при плотном контакте с токоведущей частью. Причинами так же может служить плохая изоляция токоведущих проводов в помещении. С этой целью обычно все токоведущие провода и соединительные кабели между ЭВМ и оргтехникой располагаются в местах недоступных для пользователей. При соблюдении ПТБ вероятность подвергнуться воздействию этой опасности сводится к минимуму. Для снижения вероятности реализации этой опасности также необходимо проверять изоляцию токоведущих проводов, соединений аппаратов, их исправность.

При работе с ПЭВМ работники обязаны:

**–**соблюдать режим труда и отдыха, установленный законодательством, правилами внутреннего трудового распорядка организации, трудовую дисциплину, выполнять требования охраны труда, правил личной гигиены;

**–**выполнять требования пожарной безопасности, знать порядок действий при пожаре, уметь применять первичные средства пожаротушения;

**–**курить только в специально предназначенных для курения местах;

**–**знать приемы оказания первой помощи при несчастных случаях на производстве;

**–**о неисправностях оборудования и других замечаниях по работе с ПЭВМ сообщать непосредственному руководителю или лицам, осуществляющим техническое обслуживание оборудования.

Не допускается:

**–** выполнять работу, находясь в состоянии алкогольного опьянения либо в состоянии, вызванном употреблением наркотических средств, психотропных или токсических веществ, а также распивать спиртные напитки, употреблять наркотические средства, психотропные или токсические вещества на рабочем месте или в рабочее время;

**–**устанавливать системный блок в закрытых объемах мебели, непосредственно на полу;

**–**использовать для подключения ПЭВМ розетки, удлинители, не оснащенные заземляющим контактом (шиной).

## 5.4 Электробезопасность

Основным опасным фактором при работе на компьютере является поражение электрическим током. Чтобы избежать поражения током нужно соблюдать технику безопасности при работе на компьютере, а также использовать соответствующие технические средства.

Перед началом работы следует убедиться в исправности электропроводки, выключателей, штепсельных розеток, при помощи которых оборудование включается в сеть, наличии заземления компьютера, его работоспособности.

Запрещается приступать к работе при:

**–** обнаружении неисправности оборудования;

**–** поврежденных кабелях электропитания;

**–** отсутствии защитного заземления устройств ПЭВМ;

**–**отсутствии защитного фильтра класса “полная защита” либо отключении заземляющего проводника фильтра [3].

Техническими способами и средствами защиты от поражения электрическим током являются: защитное заземление, зануление, выравнивание потенциалов, малое напряжение.

Защитное заземление – преднамеренное электрическое соединение с землёй или её эквивалентом металлических нетоковедущих частей, которые могут оказаться под напряжением. Назначение защитного заземления – устранение опасности поражения работающего электрическим током при появлении напряжения на конструктивных частях электрооборудования.

После окончания работы необходимо обесточить все средства вычислительной техники и периферийное оборудование. В случае непрерывного производственного процесса необходимо оставить включенными только необходимое оборудование [3].

## 5.5 Пожарная безопасность

Пожарная безопасность объекта – состояние объекта, при котором с регламентирую щей вероятностью исключается возможность возникновения и развития пожара, а также обеспечивается защита людей и материальных ценностей от воздействия его опасных факторов.

Пожарная безопасность может быть обеспечена мерами пожарной профилактики и активной пожарной защитой.

Для обеспечения пожаро- и взрывобезопасности каждый работающий на предприятии должен выполнять следующие требования:

**–**знать и выполнять требования правил пожарной безопасности как в целом по обществу, так и по своему рабочему месту;

**–** знать расположение средств пожаротушения на производстве и уметь ими пользоваться;

**–**знать местонахождение ближайшего телефона, пожарного извещателя и способ вызова аварийно-спасательной службы; знать и уметь выполнять свои обязанности в случае пожара, загорания или аварии.

Для предупреждения возникновения пожаров мерами пожарной профилактики в помещениях оборудованных компьютерной техникой необходимо соблюдать основные требования по обеспечению пожарной безопасности. При возникновении пожароопасной ситуации или пожара персонал должен немедленно принять необходимые меры для его ликвидации, одновременно оповестить о пожаре администрацию. Оповещение людей о пожаре должно осуществляться во все помещения здания с постоянным или временным пребыванием людей путём подачи звуковых сигналов, трансляцией речевой информации о необходимости эвакуации и других действиях, направленных на обеспечение безопасности.

Для тушения пожаров в электроустановках под напряжением наиболее эффективны углекислотные, порошковые огнетушители. В данном случае целесообразно использовать огнетушители марок ОУ-5, ОУ-8, ЧП-8М, которые применяются для тушения неинтенсивных очагов пожара [10].

В системе профилактических мер, направленных на обеспечение безопасности людей при возникновении пожара в зданиях и сооружениях, важное место занимает вопрос своевременной и организованной их эвакуации.

Кратковременность процесса эвакуации объясняется быстрым нарастанием при пожаре факторов, опасных для здоровья и жизни человека. К опасным факторам пожара для здоровья человека относят температуру среды в рабочей зоне или на уровне роста человека, снижение концентрации кислорода в помещениях до опасных величин, опасные концентрации продуктов горения и термического разложения, потерю видимости из-за задымления помещений и путей эвакуации, лучистые тепловые потоки. Все эти факторы вредно воздействуют на организм человека и при достижении определенных значений могут привести к смертельным исходам. Сильное задымление помещений и путей эвакуации приводит к потере ориентировки эвакуирующихся [12].

При возникновении пожара, люди должны покинуть здание за минимально короткое время согласно плану эвакуации, двигаясь в направлении к выходу из здания.

# Заключение

В результате дипломного проектирования по теме «Разработка системы мониторинга устройств и сервисов локальной вычислительной сети» были решены следующие задачи:

1. Выполнен сравнительный анализ и изучены возможности существующих похожих проектов, в результате чего были сформированы требования к функциональности разработанного программного продукта, выбраны средства и технологии, использованные при разработке.
2. В результате проектирования системы были выполнены следующие задачи:

* разработана концепция пользовательского интерфейса системы;
* выполнено проектирование базы данных;
* разработаны объекты базы данных;
* разработан пользовательский интерфейс;
* разработана система единой авторизации.

1. Разработана и оптимизирована система мониторинга активности устройств ЛВС которая предоставляет следующие возможности:

* авторизация пользователей;
* сбор данных об активности хостов, показателей датчиков, ресурсов сервисов;
* структурирование и запись информации в бд;
* отображение информации пользователю;
* быстрая навигация и поиск;
* отображение критических секций в реальном режиме времени;
* отображение графической информации;
* логирование действий пользователей в системе.

1. Рассчитана сметная стоимость разработанного программного продукта, рассмотрены вопросы охраны труда и техники безопасности.
2. Разработана проектная документация в составе:

* техническое задания (см. приложение А);
* описание программы (см. приложение Г);
* руководства пользователя (см. приложение Д);

Разработанный программный продукт на данный момент используется на месторождении.

# Список используемых источников

1. Концепция государственного управления охраной труда в Республике Беларусь: утв. Постановлением Совета Министров Республики Беларусь от 16 августа 2005 г. № 904 – Мн: Дикта, 2007. – 6 с.
2. Межотраслевая типовая инструкция по охране труда при работе с персональными компьютерами: утверждена Постановлением Министерства труда и социальной защиты Республики Беларусь от 30 ноября 2004 г. № 138: текст приведен по состоянию на 14 сентября 2005 г. – Мн.: Дикта, 2005. – 220с.
3. Охрана труда при работе на персональных электронно-вычислительных машинах и другой офисной технике: Практическое пособие. Дополнительный тираж / Составители: Семич В.П., Семич А.В. – Мн.: ЦОТЖ, 2005.–86с.
4. Технико-экономическое обоснование дипломных проектов /под ред. проф. В.К. Беклешова/– М.:Высшая школа, 1997. – 145с.
5. Методические указания по организации и прохождению производственных (технологических и преддипломных) практик/ УО «ПГУ»; сост. И. Б. Бураченок, С.В. Кухта, С.П. Чеботарёв, 2011. – 60с.
6. ГОСТ 19.201-78 ЕСПД. Техническое задание. Требования к содержанию и оформлению. – М.: Издательство стандартов, 1987.
7. ГОСТ 19.301-79 ЕСПД. Программа и методика испытаний. - М.: Изд-во стандартов, 1982.
8. ГОСТ 19.402-78 ЕСПД. Описание программы. - М.: Изд-во стандартов, 1982.
9. ГОСТ 19.505-79 ЕСПД. Руководство оператора. - М.: Изд-во стандартов, 1982.
10. НПБ 28-2001. Пожарная техника. Огнетушители. Требования к эксплуатации.
11. СанПиН 9-131 РБ 2000 «Гигиенические требования к видеодисплейным терминалам, персональным электронно-вычислительным машинам и организации работы».
12. ТК РБ ст. 228 «Обязательные медицинские осмотры работников некоторых категорий».
13. ТК РБ ст. 228 «Понятие охраны труда, требования по охране труда».
14. Минский городской исполнительный комитет [Электронный ресурс]/ Разработка и сопровождение КУП «Центр информационных технологий Мингорисполкома» © 2001 – 2014 – Режим доступа: http://minsk.gov.by/ru/tarif/. – Дата доступа: 13.03.2014
15. Agavi Framework [Электронный ресурс] / 2005–2014 The Agavi Project – Режим доступа: <http://www.agavi.org> – Дата доступа: 28.02.2014
16. Doctrine [Электронный ресурс] / Справочное руководство по Doctrine 2 на русском языке. – Режим доступа: http://odiszapc.ru/doctrine/. – Дата доступа: 28.02.2014
17. DeveloperWorks [Электронный ресурс]/ IBM © – Режим доступа: <http://www.ibm.com/developerworks/> – Дата доступа: 20.03.2014
18. Microsoft Developer Network [Электронный ресурс] / Официальный сайт Microsoft. – Microsoft, 2012. – Режим доступа: <http://msdn.microsoft.com/en-us/>. – Дата доступа: 13.03.2014
19. Unified Modeling Language™ (UML®)[Электронный ресурс]/ Copyright © 1997–2014 Object Management Group, Inc. – Режим доступа: http://www.uml.org/– Дата доступа: 13.03.2014.