

Федеральное государственное автономное
образовательное учреждение
высшего образования
«СИБИРСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

Институт Космических и Информационных Технологий
институт
Информационные Системы
кафедра

УТВЕРЖДАЮ
кафедрой ИС
_____ Виденин С. А.
подпись инициалы, фамилия
« _____ » _____ 2016 г.

Зав.

ДИПЛОМНЫЙ ПРОЕКТ

230201.65 Информационные системы и технологии
код и наименование специальности

Создание информационной системы для поддержания работы системного администратора компьютерной сети

Пояснительная записка

Руководитель	_____	<u>доцент</u>	<u>П. П. Дьячук</u>
	подпись, дата	должность	инициалы, фамилия
Выпускник	_____		<u>В.Н. Дедков</u>
	подпись, дата		инициалы, фамилия
Нормоконтролер	_____		<u>Л.С. Троценко</u>
	подпись, дата		инициалы, фамилия

Красноярск 2016

										Лист
										2
Изм.	Коллич.	Лист.	№ док	Подпись	Дата	ДП-230201.65-0803851 ПЗ				

ДП-230201.65-0803851 ПЗ

Лист

3

<i>Изм.</i>	<i>Коллич.</i>	<i>Лист.</i>	<i>№ док</i>	<i>Подпись</i>	<i>Дата</i>

СОДЕРЖАНИЕ

Глава 1 Общие сведения.....	8
1.1 Цели и задачи	8
1.2 Назначение разработки	8
1.3 Требования к программе и программному продукту	8
1.3.1 Требования к надежности.....	9
1.3.2 Требования к аппаратным средствам.....	9
1.4 Стадии и этапы разработки	10
1.5 Обоснование выбора СУБД MySQL	10
1.6 Обоснование выбора среды разработки Visual Studio	11
Глава 2 Анализ предметной области.....	13
2.1 Анализ существующих программных средств.....	13
2.1.1 Страйк: Инвентаризация компьютеров.....	13
2.1.2 PC Inventory Advisor 3.0.....	15
Глава 3 Проектирование системы	17
3.1 Функциональная модель системы	17
3.2 Диаграмма классов	18
3.3 Физическая и логическая модель системы	20
3.4 Структура модели данных	23
Глава 4 Программная реализация системы	30
4.1 Приложения, использованные при разработке проекта	30
4.1.1 CA ERWin Data Modeler Community Edition	30
4.1.2 IBM Rational Rose Modeler	32
4.1.3 MySQL Workbench 6.3 CE.....	33
4.1.4 Система контроля версий Mercurial	35
Глава 5 Эргономика	37
5.1 Эргономика эффективное взаимодействия интерфейса «человек-компьютер».....	37
5.2 «Usability» простота и удобство в использовании	38

Изм.	Коллич.	Лист.	№ док	Подпись	Дата

5.3	Интерфейс разработанной системы.....	39
Глава 6 Информационная безопасность		43
6.1	Перечень возможных угроз	43
6.2	Выбор методов и средств защиты информации.....	45
6.3	Создание архивов баз данных и их восстановление.....	45
ЗАКЛЮЧЕНИЕ		48
СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ		49

ВВЕДЕНИЕ

В настоящее время объём обрабатываемой информации возрастает в геометрической прогрессии и вместе с тем постоянно усложняются, и сами средства обработки информации. С усложнением программного обеспечения появляется проблема с избыточностью компонентов на графической интерфейсной части. Вследствие чего наступает момент, когда простые для исполнения действия или поиск определенной информации требуют больше времени и действий чем, хотелось бы. Например, современное программное обеспечение по сбору информации о ПЭВМ и их комплектующих, а также по помощи в составлении отчетов, становится довольно нагромождённым, что в свою очередь дает нагрузку на вычислительные способности оборудования.

Также одной из проблем для системного администратора в современном программном обеспечении для работы с информацией об оборудовании можно назвать закрытость его исходных кодов, что препятствует модификации под собственные нужды. Также всегда остаётся вероятность, что разработчик программного обеспечения, у которого исходный программный код закрыт, мог оставить некоторые уязвимости, что впоследствии может привести к печальным последствиям, например, дать доступ к информации с ограниченным доступом.

Поэтому возникает необходимость в создании собственной информационной системы, которая поможет системному администратору оперировать с необходимой информацией, а также уменьшает временные затраты на поиск и доступ к данным. Система не должна содержать в себе излишнее количество графических компонентов, упрощенный графический интерфейс должен способствовать простому использованию информационной системы. Разрабатываемая система будет являться тем продуктом, в котором дальше со временем можно дорабатывать необходимый функционал под

						ДП-230201.65-0803851 ПЗ	Лист
							6
<i>Изм.</i>	<i>Колоч.</i>	<i>Лист.</i>	<i>№ док</i>	<i>Подпись</i>	<i>Дата</i>		

собственные нужды, которые могут меняться по истечении некоторого времени.

Благодаря данной программе системный администратор сможет с наименьшими затратами времени получать доступ к необходимой информации для составления отчета по оборудованию и осуществления своей профессиональной деятельности.

В ходе работы над дипломным проектом, был сделан обзор на аналогичные программные продукты, рассмотрены их основные функции, а также исследованы существующие методы, средства и технологии для разработки информационно справочных систем, также была спроектирована логическая и физическая модель базы данных.

Основываясь на результатах проведенных исследований, была разработана информационно-справочная система для поддержания работы системного администратора компьютерной сети.

Изм.	Коллич.	Лист.	№ док	Подпись	Дата

ДП-230201.65-0803851 ПЗ

Лист

7

Глава 1 Общие сведения

1.1 Цели и задачи

Цель создания информационной системы для поддержания работы системного администратора компьютерной сети.

Задачи:

- 1) Разработать структуру базы данных для хранения информации об оборудовании и сотрудниках;
- 2) С проектировать систему взаимодействующую с системой базы данных для осуществления необходимой функциональности системы.

1.2 Назначение разработки

Разработка информационной системы предназначена для более эффективной работы системного администратора компьютерной сети и упрощения доступа к информации.

Цели создания системы:

- обеспечение системой простых и удобных механизмов для работы с базой данных;
- сбор, обработка и оперативная выдача информации.

1.3 Требования к программе и программному продукту

Программа должна выполнять все необходимые функции, а также быть пригодной для эксплуатации.

Работа с системой не должна требовать специальных знаний и высокой квалификации пользователя. Пользовательский интерфейс должен быть прост в

Изм.	Колоч.	Лист.	№ док	Подпись	Дата

понимании и освоении. Интуитивно понятная структура должна способствовать комфортной работе пользователя с системой.

Все пояснительные надписи рабочих окон, создаваемой информационной системы должны быть выполнены на русском языке. Надписи на управляющих элементах должны однозначно интерпретироваться пользователем.

Информация, заложенная в систему должна храниться в одном месте и предоставляться пользователю в соответствии с его запросами.

Информационная система для поддержания работы системного администратора компьютерной сети должна реализовывать возможность по добавлению, редактированию и удалению записей из таблиц БД системы.

1.3.1 Требования к надежности

Программа должна быть устойчивой к сбоям внутри самой программы и при воздействии внешних условий. Каждый конкретный элемент системы должен быть снабжен механизмом защиты от некорректных действий пользователя.

1.3.2 Требования к аппаратным средствам

Разработанный продукт должен функционировать при минимальных требованиях к аппаратному обеспечению системы:

- процессор: Intel или совместимый с частотой 2 GHz и более;
- ОЗУ: 1024 MB;
- VGA адаптер;
- LAN (если база данных находится в локальной сети);
- Internet (если база данных находится в глобальной сети);
- операционная система Windows XP SP1 или выше.

Изм.	Коллич.	Лист.	№ док	Подпись	Дата

1.4 Стадии и этапы разработки

Этап 1. Постановка задачи.

Этап заключался в ознакомлении с предметной областью и изучением документации.

Этап 2. Анализ требований.

Данный этап включает в себя анализ аналогичного программного обеспечения для выявления положительных и отрицательных сторон, на основании чего должен быть спроектирован и разработан собственный проект.

Этап 3. Проектирование системы

На этапе проектирования системы должны быть проведены работы по разработке логической модели данных и работы по моделированию функциональных требований к разрабатываемой системе.

По окончании работ, должна быть осуществлена деятельность по выбору программных средств решения поставленных задач, описаны структуры входных и выходных данных, разработана структура и интерфейс программы.

Этап 4. Реализация системы.

На данном этапе должен быть реализован интерфейс программного продукта, а также заполнены справочники, необходимые для работы системы.

1.5 Обоснование выбора СУБД MySQL

Выбор в качестве системы управления базами данных СУБД MySQL

При сравнении производительности на несложных запросах, при количестве клиентов не более десяти, MySQL уверенно лидирует по сравнению с MSSQL и Oracle. При более больших нагрузках и сложных запросах MySQL теряет свои позиции. В случае информационного киоска, количество одновременных клиентов не будет превышать пять человек. Следовательно, в данном случае MySQL является оптимальным вариантом.

Изм.	Колоч.	Лист.	№ док	Подпись	Дата

ДП-230201.65-0803851 ПЗ

Лист

10

Выбор данной СУБД обусловлен следующими ее характеристиками:

- MySQL является клиент-серверной СУБД;
- MySQL является реляционной базой данных;
- MySQL – свободно распространяемый продукт;
- MySQL является кросс-платформенным продуктом.

1.6 Обоснование выбора среды разработки Visual Studio

Microsoft Visual Studio - продукт компании Майкрософт, имеющий встроенную среду разработки ПО и ряд других инструментальных средств. Продукт позволяет разрабатывать консольные приложения и приложения с визуальным интерфейсом, в том числе с поддержкой технологии Windows Forms, а также веб-сайты, веб-службы и веб-приложения, как в собственном, так и в управляемом кодах для всех платформ, поддерживаемых компанией Microsoft.

Visual Studio включает в свой состав редактор исходного кода с поддержкой технологии IntelliSense и возможностью примитивного рефакторинга кода. Встроенный отладчик среды разработки работает, как на уровне исходного, так и на уровне машинного кода. Остальные встраиваемые инструменты включают в себя редактор форм для упрощения разработки графического интерфейса приложения, дизайнер схемы и классов баз данных веб-редактор. Visual Studio позволяет использовать в своей работе сторонние дополнения для расширения функциональности на каждом уровне, включая добавление поддержки систем контроля версий разрабатываемого кода, добавление новых наборов инструментов.

C# - объектно-ориентированный язык программирования. Разработан в 1998-2001 годах группой инженеров под руководством Андерса Хейлсберга в компании Microsoft, как язык разработки приложений для платформы Microsoft .NET Framework.

Изм.	Копич.	Лист.	№ док	Подпись	Дата

ДП-230201.65-0803851 ПЗ

Лист

11

Основные преимущества языка C#:

- объектно-ориентирован, поддерживает перегрузку операторов (в том числе операторов явного и неявного типа приведения), полиморфизм, события, делегаты, свойства, атрибуты, обобщённые методы и типы;
- простота в использовании, язык относится к семье языков C и обладает подобным синтаксисом, из них его синтаксис наиболее близок к C++ и Java;
- поддержка Visual C# в Visual Studio реализована в виде полноценного редактора кода, шаблонов проектов, компилятора, мастеров кода, конструкторов, удобного и мощного отладчика и многих других средств. Библиотека классов .NET Framework дает доступ ко многим службам ОС и к другим полезным классам, включенным в установочный пакет, что существенно ускоряет процесс разработки;
- кроссплатформенность, благодаря проектам Portable.NET и Mono, позволяющие запускать программы .NET, в том числе и программы написанные на C# на других ОС;
- простота подключения к БД, редактирования и представления структур и данных таблиц с помощью встроенных библиотек.
- Одной из основных причин в выборе C# и технологии .Net это библиотека MySQL Connector/Net. В отличии от аналогичной библиотеки от c++ для которой была необходима библиотека boost.

Глава 2 Анализ предметной области

2.1 Анализ существующих программных средств

Информационная система должна способствовать сокращению затрачиваемого времени на составление технических отчетов, а также обеспечивать простой и быстрый доступ к необходимой информации по администрированию и решению возникающих проблем.

Система должна соответствовать следующим требованиям:

- ± Надежность;
- ± Низкая нагрузка на вычислительные ресурсы.

Анализ существующего ПО позволит выявить их достоинства и недостатки, а также применить полученную информацию при создании собственного проекта.

2.1.1 Страйк: Инвентаризация компьютеров

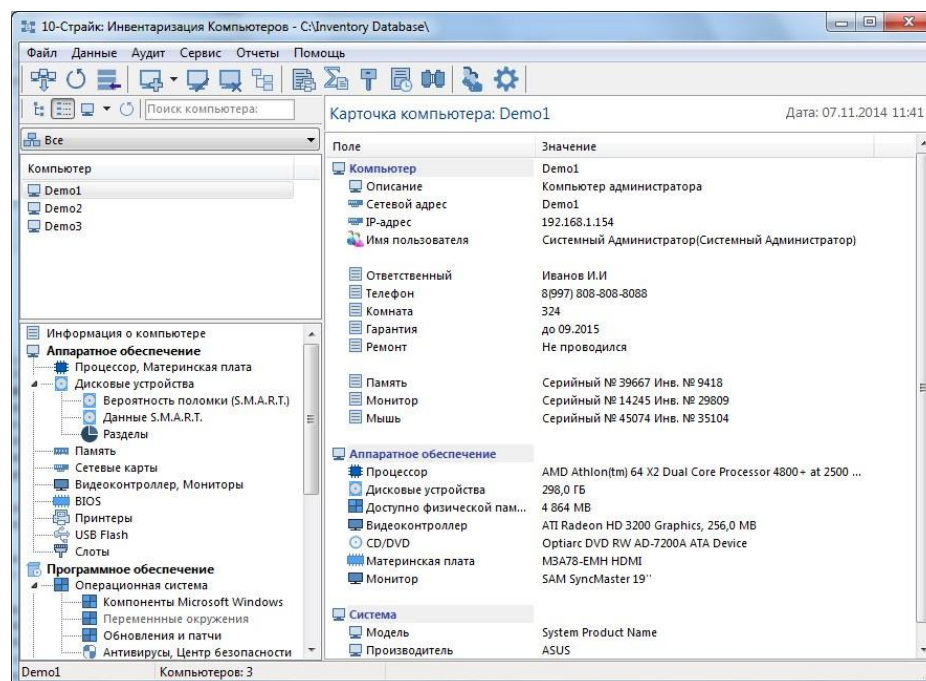


Рисунок 2.1 – «10-Страйк: Инвентаризация Компьютеров»

Изм.	Колич.	Лист.	№ док	Подпись	Дата

«10-Страйк: Инвентаризация Компьютеров» – программное решение для выполнения задач в области учета и инвентаризации компьютеров на предприятие, позволяет администраторам сетей создать и администрировать базу данных программ, лицензий, комплектующих и компьютеров. Дает возможность отслеживать конфигурации удаленных компьютеров посредством сетевых средств, вести учет программного и аппаратного обеспечения на них.

Осуществив сбор информации в инвентарную базу данных, можно получить подробную справку о периферии, имеющихся в наличии компьютеров. Типы установленных процессоров, объем оперативной памяти, типы и количество жестких дисков, информацию о созданных разделах на дисках, наличии установленных приводов, использующихся USB-устройствах, подключенных принтерах и прочей периферии, установленной ОС, приложениях и лицензиях, программах в автозагрузке и так далее. Средствами рассматриваемого ПО можно осуществлять учет всех используемых программных и аппаратных средств сети.

Основные возможности ПО «10-Страйк: Инвентаризация Компьютеров»:

- учет и инвентаризация используемого аппаратного обеспечения; формирование отчетов по установленным аппаратным средствам на компьютерах сети, мониторинг изменений в конфигурациях компьютеров, отслеживание состояния жестких дисков;
- сопровождение и инвентаризация установленного ПО и лицензий. Формирование отчетов: по наличию и количеству определенных программ на компьютерах, версиям ОС, установленным обновлениям, наличию определенных приложений в автозагрузке, установленных кодеках и драйверах;
- формирование профильных отчетов по конкретному аппаратному обеспечению и используемым программам. Сбор данных о программном и аппаратном обеспечении компьютеров организации в режиме реального времени;

- Планирование и подготовка обновлений компьютеров и программного обеспечения.

2.1.2 PC Inventory Advisor 3.0

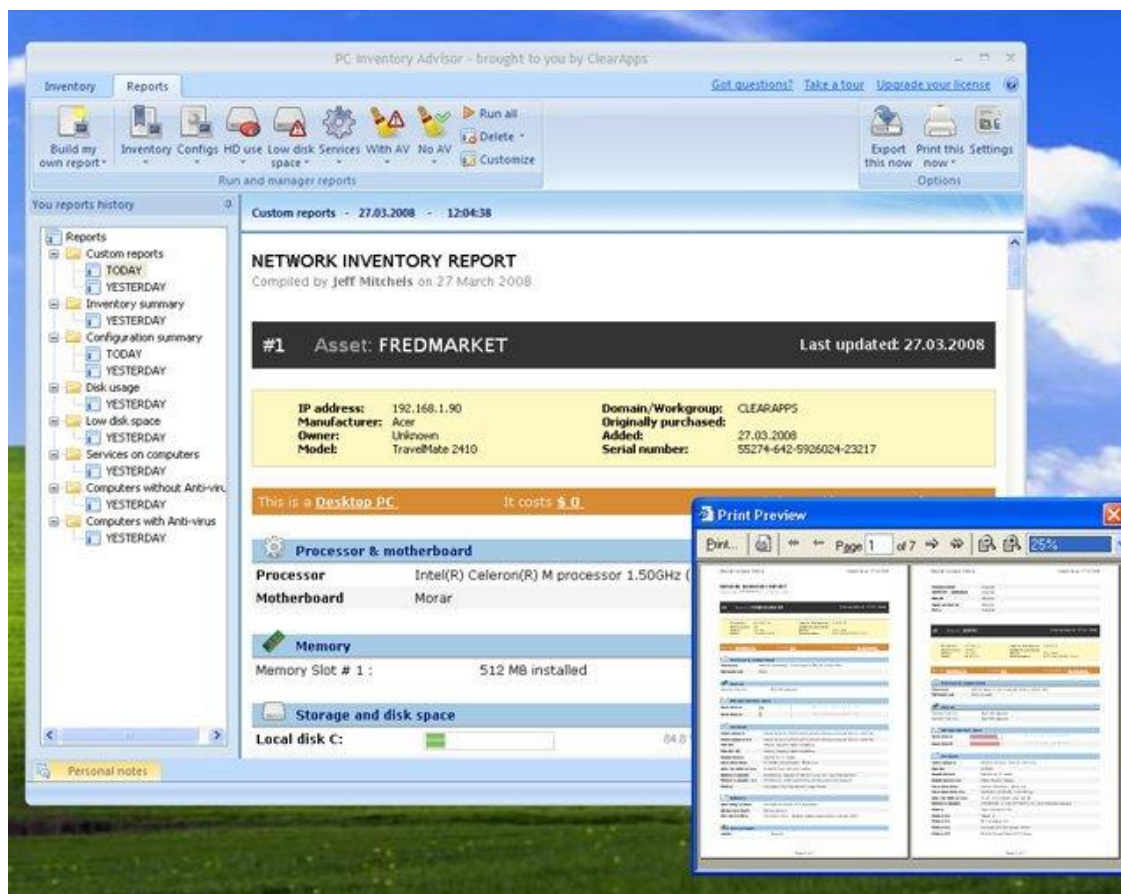


Рисунок 2.1 – «PC Inventory advisor 3.0»

«PC Inventory Advisor 3.0» - это решение для сетевой инвентаризации компьютеров, находящихся в сетей Windows, с отчетами, предоставляемыми в формате HTML. Программное обеспечение позволяет осуществлять учет программ и аппаратуры, затрачивая на это намного меньше ресурсов. Программный продукт в автоматическом режиме формирует список сетей любого размера и типов устройств, которыми располагает пользователь.

Рассматриваемое ПО позволяет определять, какая программа установлена на конкретном компьютере сети. Кроме того позволяет отслеживать

Изм.	Коллич.	Лист.	№ док	Подпись	Дата

деятельность установленных корпоративных программ, а так же программам, обеспечивающих безопасность сети и управления лицензиями.

Программа PC Inventory Advisor дает возможность пользователю избежать потери оборудования и спланировать сетевые обновления на основании полученных данных. ПО предоставляет возможность получать в реальном времени информацию о понижении производительности рабочих станций, количество свободного места на жестком диске, остаток краски в принтере. Имеется возможность добавить его в список сетевой инвентаризации и добавить в свои отчеты.

PC Inventory Advisor автоматически анализирует все данные, касающиеся оборудования и своевременно уведомляет, в случае критических событий, предоставляя вам возможность настраивать сетевые уведомления для большего удобства.

Изм.	Колоч.	Лист.	№ док	Подпись	Дата

ДП-230201.65-0803851 ПЗ

Лист

16

Глава 3 Проектирование системы

3.1 Функциональная модель системы

Передовым подходом в разработки программного обеспечения является объектно-ориентированный. Основным элементом в данном подходе выступает класс или объект. В объектно-ориентированном подходе за основу принят принцип сборки системы из отдельных компонентов. Средство Rational Rose - CASE разработанное фирмой Rational Software Corporation - предназначено для автоматизации этапов анализа и проектирования ПО, а также для генерации кода на различных языках и выпуска проектной документации.

В ходе работы специалистов компании Rational Software Corporation была разработана универсальная нотация для моделирования объектов (Unified Modeling Language), которая претендует на роль стандарта в области объектно-ориентированной проектной работы и анализа.

Основные задачи, которые выполняет Rational Rose, заключаются в построение различного рода спецификаций и диаграмм, определяющих физическую и логическую структуру модели, ее динамические и статические аспекты. Такие как диаграммы состояний, классов, сценариев, процессов, модулей и т.д.

Функциональное назначение системы и основные задачи, которые система должна делать, описывают диаграммы вариантов использования.

Основной смысл диаграммы вариантов использования заключается в том, что разрабатываемая система представляется в виде большого количества актеров или сущностей, взаимодействующих с системой с помощью различных вариантов использования. При этом действующим лицом или актером (actor) называется любая сущность, взаимодействующая с системой и не находящаяся в ней. Это может быть техническое устройство, человек, программа или любая другая система, которая может воздействовать на моделируемую систему так,

Изм.	Коллич.	Лист.	№ док	Подпись	Дата

как определит ее разработчик. Вариант использования предназначен для описания сервисов, к которым актер получает доступ от системы.

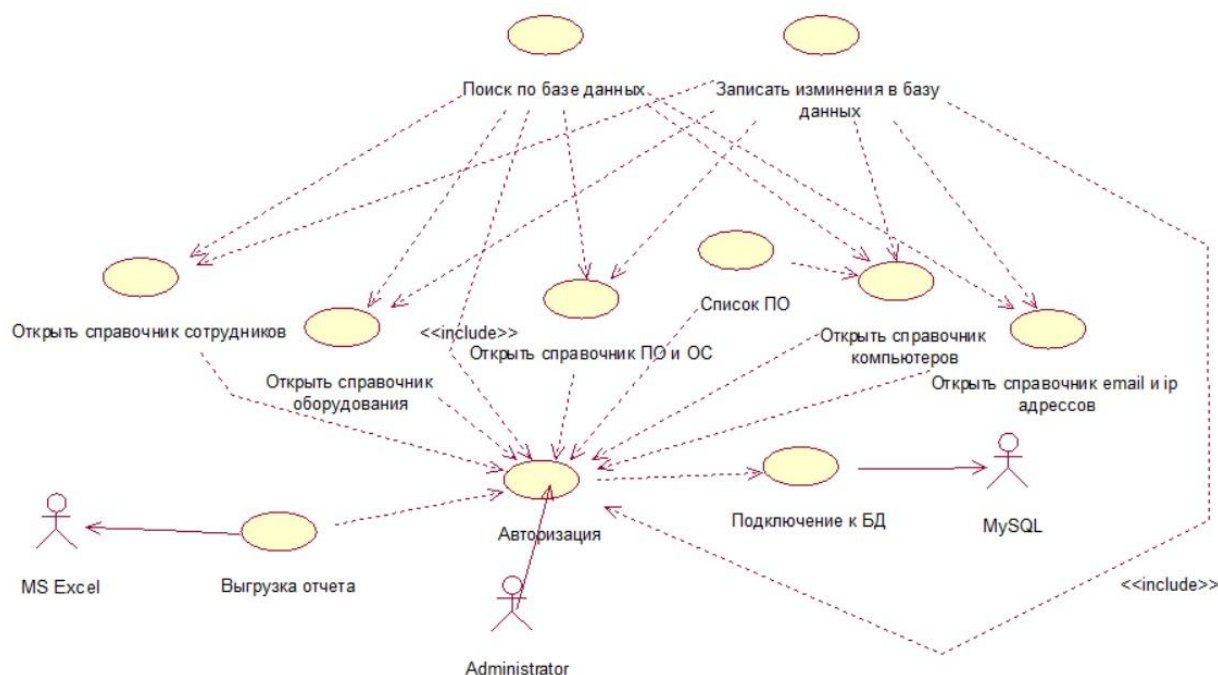


Рисунок 3.1 – Диаграмма вариантов использования

3.2 Диаграмма классов

Одной из основных диаграмм при разработке программного обеспечения методом ООП является диаграмма классов. Нотация классов в языке UML проста в понимании абсолютно для всех, кто когда-либо имел опыт работы с CASE-инструментариями. Подобная нотация применяется и для экземпляров класса - объектов, с небольшим отлитием, которое заключается в том, что вся надпись подчеркивается и к имени класса добавляется имя объекта.

Предназначением диаграммы классов является представление статической структуры системы в терминологии классов ООП. В диаграмме классов могут быть отражены разнообразные взаимосвязи между отдельными сущностями предметной области, такими как подсистемы и объекты, а также описывает их типы отношений и внутреннюю структуру. На диаграмме не отображается информация о временных аспектах функционирования системы.

Изм.	Колоч.	Лист.	№ док	Подпись	Дата

С данной точки зрения диаграмма классов является продолжением концептуальной модели проектируемой системы.

На рисунках 3.2 и 3.3 представлены диаграммы классов, отражающие взаимосвязь между формами и классам отвечающими за аутентификацию и соединение с базой данных.

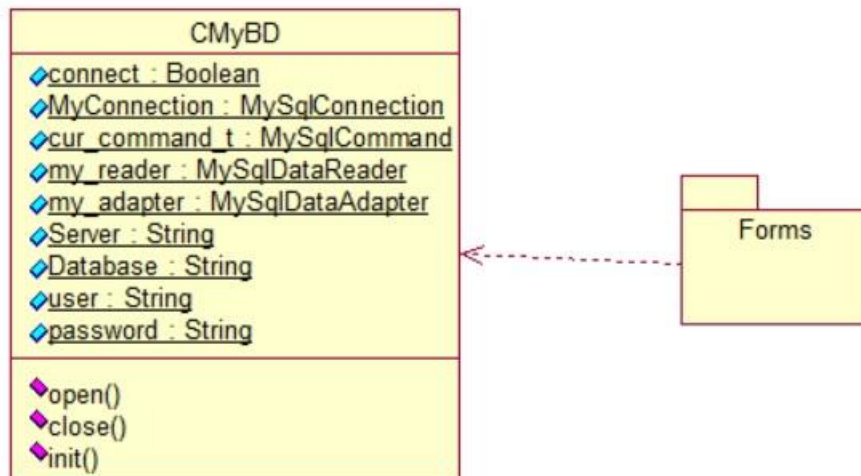


Рисунок 3.2 – Диаграмма классов

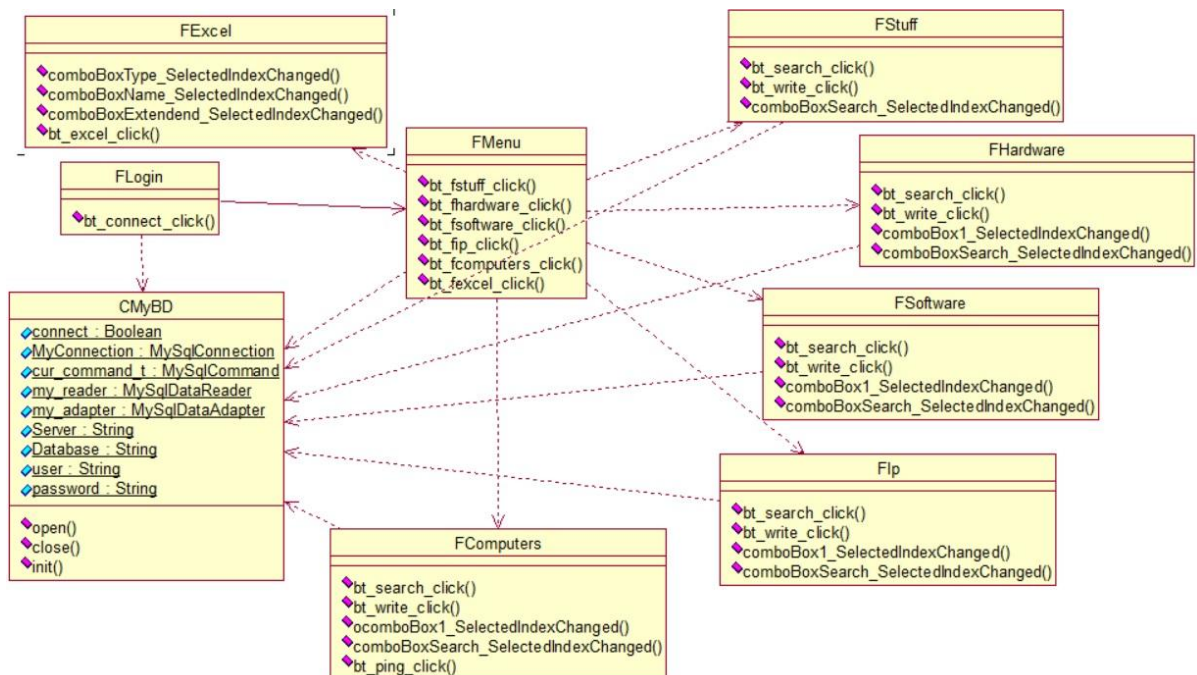


Рисунок 3.3 – Диаграмма классов формы

3.3 Физическая и логическая модель системы

Для проектирования базы данных (БД) выбрано CASE-средство MySQL Workbench, имеющее доступный и наглядный интерфейс, простотой в освоение и использовании. MySQL Workbench является наиболее распространенной и доступной, на данный момент, программой для проектирования баз данных MySQL.

Функции и структура: семейство продуктов MySQL Workbench представляет собой набор средств концептуального моделирования данных, основывающихся на методе IDEF1X. MySQL Workbench позволяет реализовать проектирование схемы базы данных, реверсный инжиниринг существующей БД и генерацию ее описания на целевом языке СУБД MySQL.

Логический уровень – взгляд на данные абстрактным образом, на нем данные называются так, как они названы в реальном мире и представляются так, как выглядят в реальном мире. Объекты и модели, представляемые на логическом уровне, называются атрибутами и сущностями. Сущность – может представлять собой место, человека, вещь, концепцию, событие о котором хранится информация. Между сущностями существует связь представляющая собой их функциональную зависимость. У каждой связи имеется атрибут. Атрибут - это свойство объекта, характеризующее его экземпляр. Логическая модель данных является универсальной и никак не связана с определенной реализацией СУБД. С другой стороны физическая модель данных, являясь отображением системного каталога, зависит от определенной СУБД.

Физическая модель включает в себя информацию по объектам БД. Вследствие того, что стандарты на объекты БД не определены, физическая модель зависит от конкретного исполнения СУБД. Следовательно, одной и той же логической модели могут соответствовать несколько разных физических моделей. В физической модели важно описать всю информацию о конкретных физических объектах – колонках, индексах, таблицах, процедурах и т.д. в то

Изм.	Коллич.	Лист.	№ док	Подпись	Дата

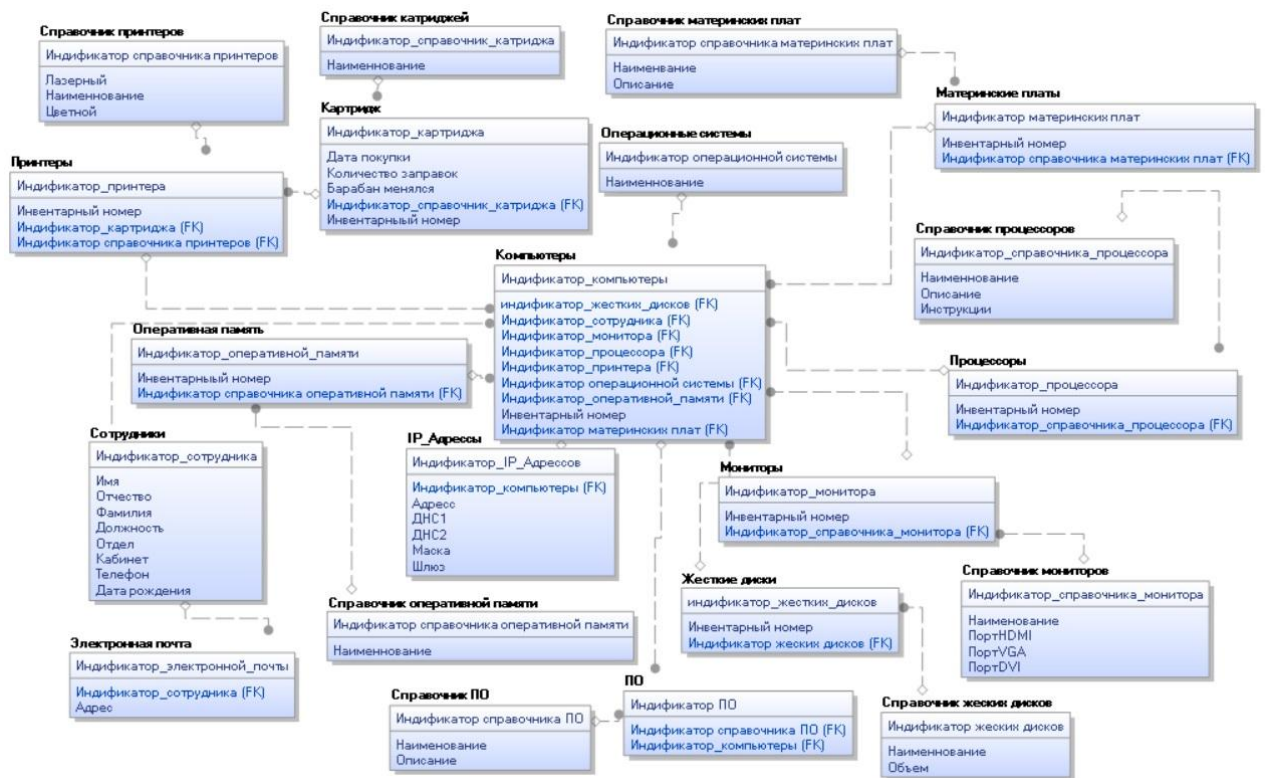


Рисунок 3.4 – Логическая схема базы данных

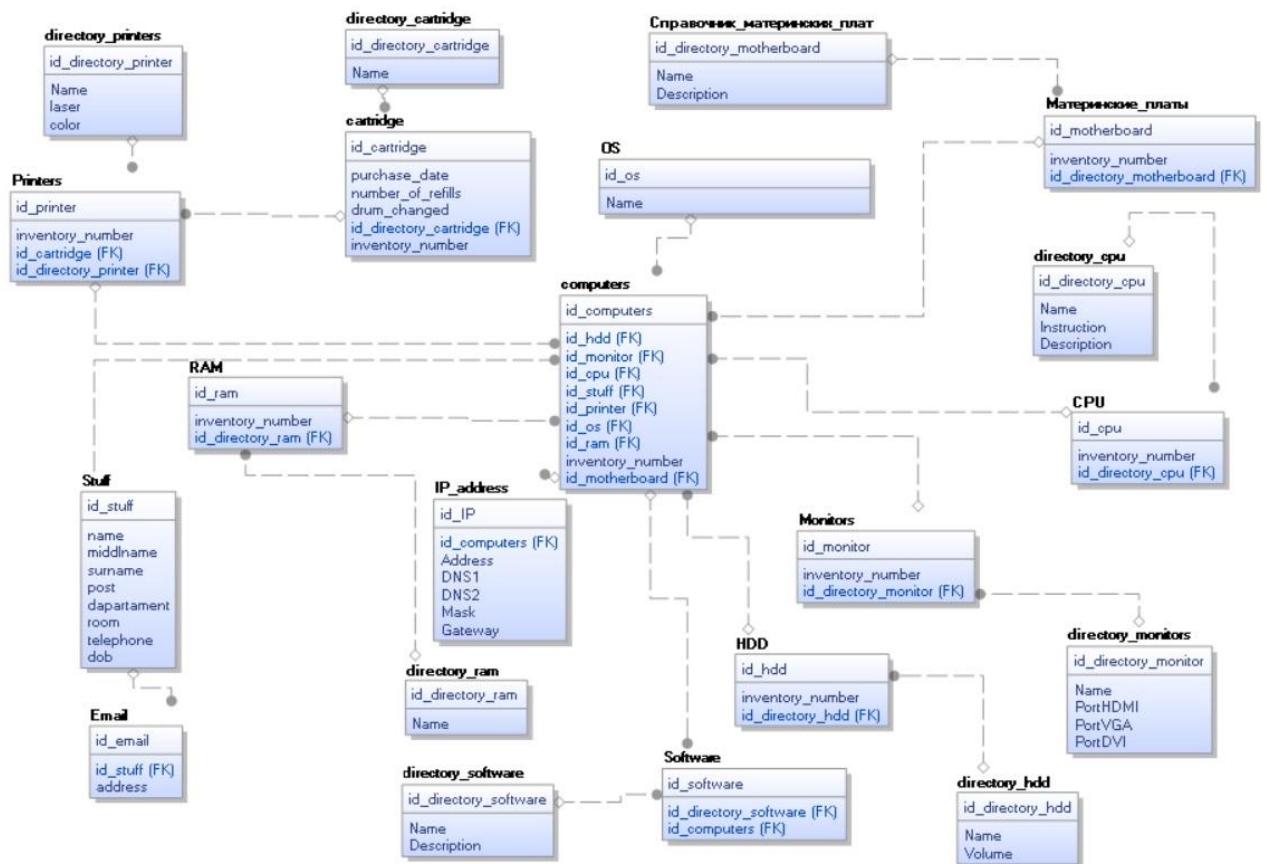


Рисунок 3.5 – Физическая схема базы данных

Изм.	Колоч.	Лист.	№ док	Подпись	Дата

3.4 Структура модели данных

Рассмотрим таблицы, отвечающие за справочники:

- Справочник по картриджам (таблица Directory_cartridge, структура представлена в таблице 3.1)
- Справочник оперативной памяти (таблица Directory_ram, структура представлена в таблице 3.2)
- Справочник по процессорам (таблица Directory_cpu, структура представлена в таблице 3.3)
- Справочник по мониторам (таблица Directory_monitors, структура представлена в таблице 3.4)
- Справочник жестких дисков (таблица Directory_hdd, структура представлена в таблице 3.5)
- Справочник ПО (таблица Directory_software, структура представлена в таблице 3.6)
- Справочник материнских плат (таблица Directory_motherboard, структура представлена в таблице 3.7)
- Справочник принтеров (таблица Directory_printers, структура представлена в таблице 3.8)

Таблица 3.1 - Структура таблицы «Directory_cartridge»

Наименование	Тип поля	Назначения
id_directory_cartridge	Number	Идентификатор
Name	Varchar(20)	Наименование

Таблица 3.2 - Структура таблицы «Directory_ram»

Наименование	Тип поля	Назначения
id_directory_ram	Number	Идентификатор
Name	Varchar(20)	Наименование

Таблица 3.3 - Структура таблицы «Directory_cpu»

Наименование	Тип поля	Назначения
id_directory_cpu	Number	Идентификатор
Name	Varchar(20)	Наименование
Instruction	Varchar(100)	Набор инструкций
Description	Varchar(255)	Описание

Таблица 3.4 - Структура таблицы «Directory_monitors»

Наименование	Тип поля	Назначения
id_directory_monitors	Number	Идентификатор
Name	Varchar(20)	Наименование
PortHDMI	bool	Есть ли порт HDMI
PortVGA	bool	Есть ли порт VGA
PortDVI	bool	Есть ли порт DVI

Таблица 3.5 - Структура таблицы «Directory_hdd»

Наименование	Тип поля	Назначения
id_directory_hdd	Number	Идентификатор
Name	Varchar(20)	Наименование
Volume	Varchar(20)	Объем (Gb, Mb, Tb)

Таблица 3.6 - Структура таблицы «Directory_softwar»

Наименование	Тип поля	Назначения
id_directory_software	Number	Идентификатор
Name	Varchar(20)	Наименование
Description	Varchar(255)	Описание

Таблица 3.7 - Структура таблицы «Directory_motherboard»

Наименование	Тип поля	Назначения
id_directory_motherboard	Number	Идентификатор
Name	Varchar(40)	Наименование
Description	Varchar(255)	Описание

Изм.	Коллич.	Лист.	№ док	Подпись	Дата

Таблица 3.8 - Структура таблицы «Directory_ printers»

Наименование	Тип поля	Назначения
id_directory_printers	Number	Идентификатор
Name	Varchar(20)	Наименование
Laser	bool	Это лазерный принтер
Color	bool	Это цветной принтер

Таблицы отвечающие за списки оборудования

- Картриджи (таблица Cartridge, структура представлена в таблице 3.9)
- Процессоры (таблица Cpu, структура представлена в таблице 3.10)
- Мониторы (таблица Monitors, структура представлена в таблице 3.11)
- Жесткие диски (таблица Hdd, структура представлена в таблице 3.12)
- Оперативная память (таблица Ram, структура представлены в таблице 3.13)
- Материнская плата (таблица Motherboard, структура представлена в таблице 3.14)
- Принтеры (таблица Printers, структура представлена в таблице 3.15)

Таблица 3.9 - Структура таблицы «Cartridge»

Наименование	Тип поля	Назначения
id_cartridge	Number	Идентификатор
inventory_number	Varchar(20)	Инвентарный номер
id_directory_cartridge	Number	Идентификатор из справочника по картриджам
purchase_date	Date	Дата покупки
number_of_refills	Byte	Количество заправок
drum_changed	bool	Менялся ли барабан

Таблица 3.10 - Структура таблицы «Cpu»

Наименование	Тип поля	Назначения
id_cpu	Number	Идентификатор
inventory_number	Varchar(20)	Инвентарный номер
id_directory_cpu	Number	Идентификатор из справочника по процессорам

Таблица 3.11 - Структура таблицы «Monitors»

Наименование	Тип поля	Назначения
id_monitors	Number	Идентификатор
inventory_number	Varchar(20)	Инвентарный номер
id_directory_monitors	Number	Идентификатор по мониторам

Таблица 3.12 - Структура таблицы «Hdd»

Наименование	Тип поля	Назначения
id_hdd	Number	Идентификатор
inventory_number	Varchar(20)	Инвентарный номер
id_directory_hdd	Number	Идентификатор из справочника жестких дисков

Таблица 3.13 - Структура таблицы «Ram»

Наименование	Тип поля	Назначения
id_ram	Number	Идентификатор
inventory_number	Varchar(20)	Инвентарный номер
id_directory_ram	Number	Идентификатор оперативной памяти

Таблица 3.14 - Структура таблицы «Motherboard»

Наименование	Тип поля	Назначения
id_motherboard	Number	Идентификатор
inventory_number	Varchar(20)	Инвентарный номер
id_directory_motherboard	Number	Идентификатор из справочника материнских плат

Изм.	Колоч.	Лист.	№ док	Подпись	Дата

Таблица 3.15 - Структура таблицы «Printers»

Наименование	Тип поля	Назначения
id_printers	Number	Идентификатор
inventory_number	Varchar(20)	Инвентарный номер
id_directory_printers	Number	Идентификатор из справочника по принтерам
id_cartridge	Number	Идентификатор картриджа

Рассмотрим оставшиеся таблицы:

- Сотрудники (таблица Stuff, структура представлены в таблице 3.16)
- Компьютеры (таблица Computers, структура представлены в таблице 3.17)
- Программное обеспечение (таблица Software, структура представлены в таблице 3.18)
- Операционные системы (таблица Os, структура представлены в таблице 3.19)
- IP Адреса (таблица Ip_address, структура представлены в таблице 3.20)
- Почта (таблица Email, структура представлена в таблице 3.21)

Таблица 3.16 - Структура таблицы «Stuff»

Наименование	Тип поля	Назначения
id_stuff	Number	Идентификатор
Name	Varchar(20)	Имя
Middlename	Varchar(20)	Отчество
Surname	Varchar(20)	Фамилия
Post	Varchar(60)	Должность
Department	Varchar(40)	Отдел
Room	Varchar(20)	Кабинет
Telephone	Varchar(20)	Телефон
Dob	Varchar(20)	Дата рождения

Изм.	Колоч.	Лист.	№ док	Подпись	Дата

ДП-230201.65-0803851 ПЗ

Лист

27

Таблица 3.17 - Структура таблицы «Computers»

Наименование	Тип поля	Назначения
id_computers	Number	Идентификатор
Id_hdd	Number	Идентификатор
Id_monitor	Number	Идентификатор
Id_cpu	Number	Идентификатор
Id_stuff	Number	Идентификатор
Id_printer	Number	Идентификатор
Id_os	Number	Идентификатор
Id_ram	Number	Идентификатор
Id_motherboard	Number	Идентификатор
inventory_number	Varchar(20)	Инвентарный номер

Таблица 3.18 - Структура таблицы «Software»

Наименование	Тип поля	Назначения
id_software	Number	Идентификатор
id_directory_software	Number	Идентификатор из справочника ПО
Id_computers	Number	Идентификатор компьютера

Таблица 3.19 - Структура таблицы «Os»

Наименование	Тип поля	Назначения
id_os	Number	Идентификатор
Name	Varchar(40)	Наименование

Таблица 3.20 - Структура таблицы «Ip_address»

Наименование	Тип поля	Назначения
id_ip	Number	Идентификатор
Id_computers	Number	Идентификатор компьютера
Address	Varchar(15)	IP Адресс
DNS1	Varchar(15)	Домен имен
DNS2	Varchar(15)	Домен имен
MASK	Varchar(15)	Маска подсети

Изм.	Коллич.	Лист.	№ док	Подпись	Дата

Таблица 3.21 - Структура таблицы «Email»

Наименование	Тип поля	Назначения
id_email	Number	Идентификатор
id_stuff	Number	Идентификатор сотрудника
Address	Varchar(20)	Адрес электронной почты

Изм.	Коллич.	Лист.	№ док	Подпись	Дата

ДП-230201.65-0803851 ПЗ

Лист

29

Глава 4 Программная реализация системы

4.1 Приложения, использованные при разработке проекта

Для ускорения, оптимизации, а также повышения удобства разработки проекта были использованы следующие приложения:

- ± CA ERWin Data Modeler Community Edition
- ± IBM Rational Rose Modeler
- ± MySQL Workbench 6.3 CE
- ± Система контроля версий Mercurial

4.1.1 CA ERWin Data Modeler Community Edition

ERwin Data Modeler – CASE–средство для документирования и проектирования баз данных, которое позволяет создавать, сопровождать и документировать витрины данных, базы данных и хранилища. Модели данных помогают визуализировать структуру данных, обеспечивая эффективный процесс администрирования, организации и управления такими аспектами деятельности предприятия, как среды развертывания, уровень сложности данных и технологий баз данных.

ERwin Data Modeler предназначен для использования всеми компаниями, разрабатывающими и использующими базы данных, системными аналитиками, администраторами баз данных, проектировщиками баз данных, руководителями проектов, разработчиками. ERwin Data Modeler дает возможность управлять данными в случае корпоративных изменений, а также в условиях быстро изменяющихся технологий.

ERwin Data Modeler дает возможность графического отображения сложных структур данных. Графическая среда ERwin Data Modeler удобна в применении и значительно упрощает разработку баз данных, автоматизирует множество сложных задач, уменьшая сроки реализации

						<i>ДП-230201.65-0803851 ПЗ</i>		<i>Лист</i>
								30
<i>Изм.</i>	<i>Колпч.</i>	<i>Лист.</i>	<i>№ док</i>	<i>Подпись</i>	<i>Дата</i>			

высокопроизводительных и высококачественных транзакционных хранилищ и БД. Данное решение улучшает связь организации, обеспечивая координацию разработчиков и администраторов баз данных, многократное использование одной модели, а также наглядное представление активов данных в удобном для обслуживания и понимания формате.

Для использования был выбран продукт под лицензией Community Edition так как он бесплатный и не имеет ограничений в коммерческом использовании. Ограничения установлены только на количество моделей объектов.

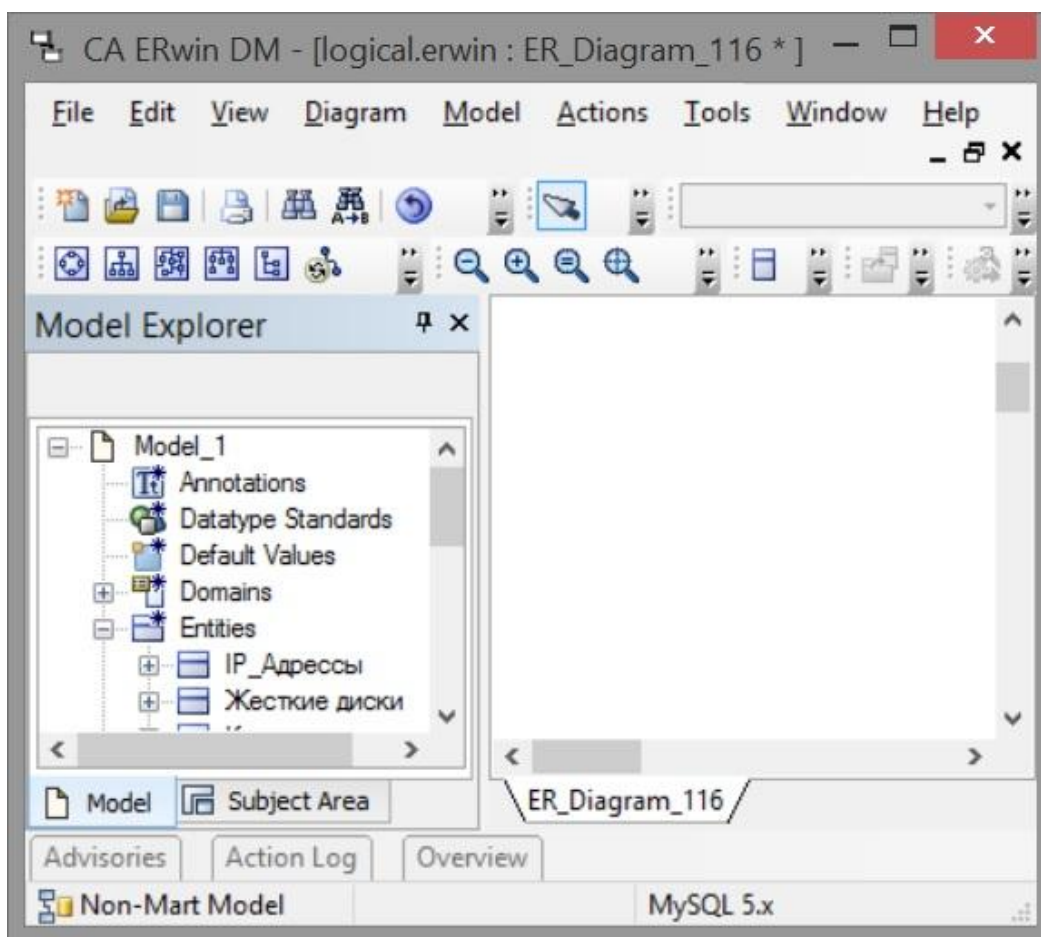


Рисунок 4.1 - «CA ERWin Data Modeler»

Изм.	Колич.	Лист.	№ док	Подпись	Дата

4.1.2 IBM Rational Rose Modeler

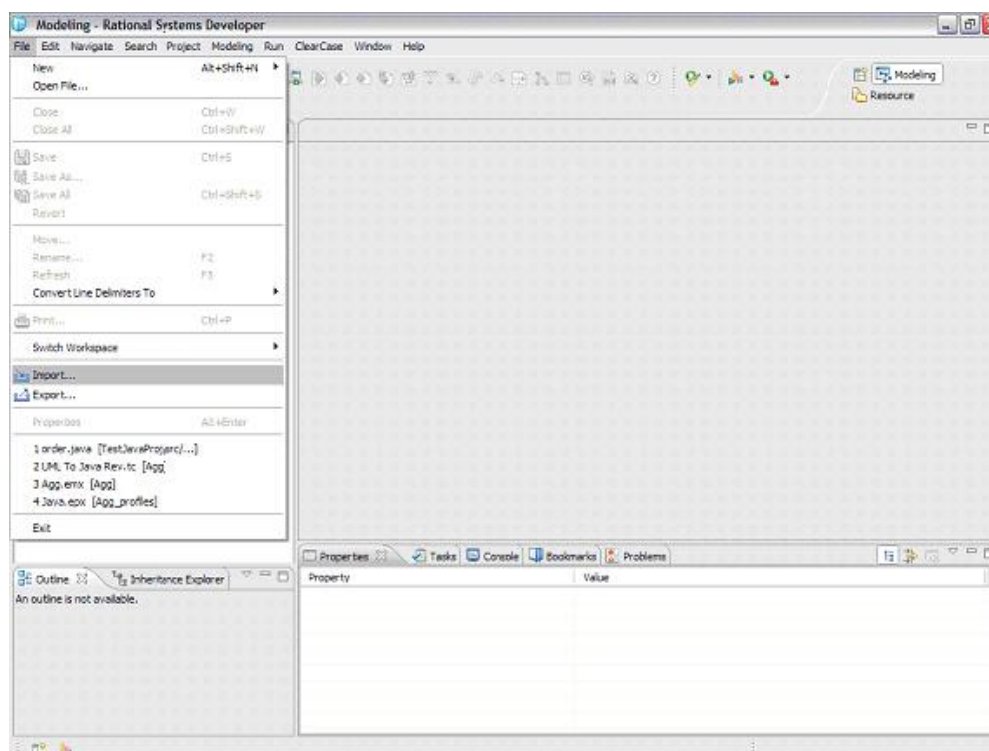


Рисунок 4.2 - «IBM Rational Rose»

IBM Rational Rose Data Modeler это программное решение для разработки приложений баз данных с использованием среды визуального моделирования. Этот продукт ускоряет процессы разработки, благодаря тому, что участники проекта и проектировщики баз данных получают единое средство моделирования с поддержкой языка UML.

С применением средства визуального моделирования Rational Rose Data Modeler аналитики, разработчики, архитекторы и проектировщики получили возможность совместной разработки приложения баз данных с учетом установленных требований:

- наглядное отображение информации - для более продуктивной разработки приложений БД необходима поддержка языка визуального моделирования;
- привязка структуры БД к логике приложения - визуальное отображение архитектуры БД и ее связи с разрабатываемым приложением;

Изм.	Колоч.	Лист.	№ док	Подпись	Дата

ДП-230201.65-0803851 ПЗ

Лист

32

- интеграция процессов проектирования БД и разработка приложения
- предоставление всем участникам разработки доступ к единой UML-модели.

4.1.3 MySQL Workbench 6.3 CE

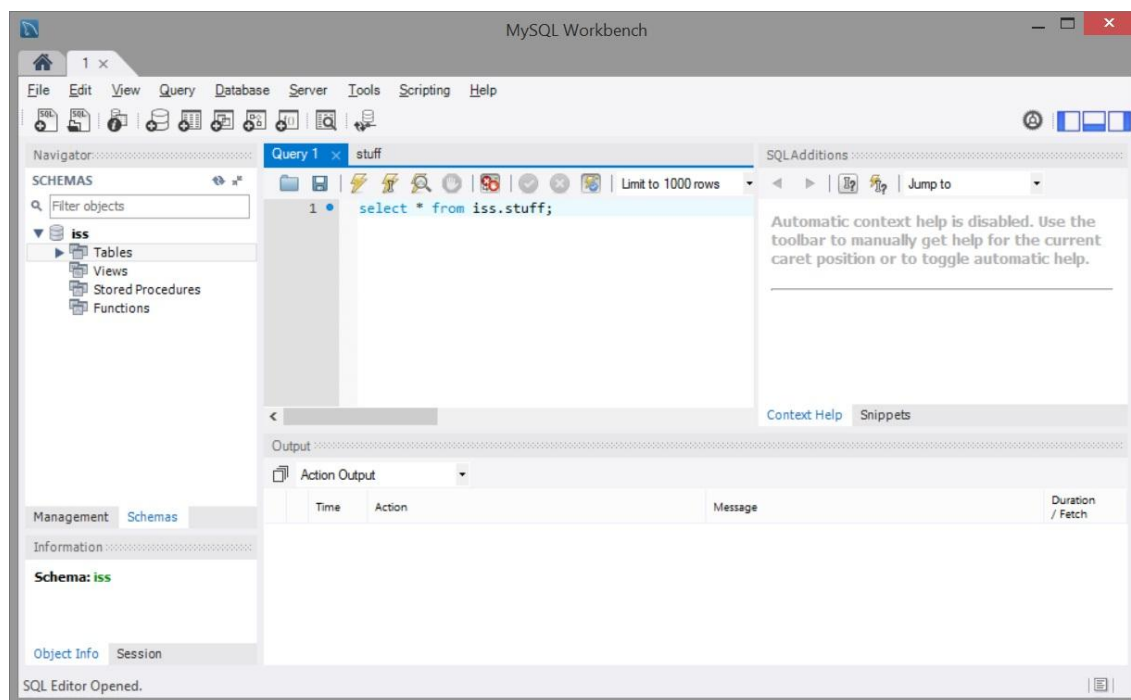


Рисунок 4.3 - «MySQL Workbench 6.3 CE»

MySQL Workbench – программное решение для визуального проектирования баз данных, интегрирующий моделирование, проектирование, эксплуатацию и создание баз данных в единое окружение для системы БД MySQL. Workbench предоставляет большой набор инструментов для конфигурирования сервера, администрирования списка пользователей и других операций. Программа доступна для работы со всеми популярными операционными системами.

MySQL Workbench дает возможность проектировщику или администратору баз данных визуально, создавать, моделировать и управлять БД. Программное обеспечение обладает всем необходимым инструментарием для создания комплексной ER-моделей, позволяет упростить процедуру сложных изменений в базе данных или вносить исправления в техническую

Изм.	Колоч.	Лист.	№ док	Подпись	Дата

документацию, в большинстве случаев эти операции занимают большое количество времени и усилий.

Программа позволяет гибко управлять SQL-запросами, создавать, выполнять и оптимизировать, используя для этого визуальный интерфейс. SQL редактор имеет удобную синтаксическую подсветку и историю выполненных SQL-запросов. Для удобства работы с несколькими базами данных одновременно реализована панель подключения к нескольким базам данных, что даёт возможность легко переключаться между ними в процессе работы.

MySQL Workbench упрощает поддержку и разработку БД, автоматизирует выполнение наиболее сложных и долгих задач, а так же улучшает взаимодействие между администраторами и разработчиками баз данных. Продукт даёт возможность проектировщикам данных отобразить требования графически и быстро решить проблему, не затрачивая на это большое количество трудовых ресурсов. При помощи MySQL Workbench можно создавать надёжные и хорошо структурированные БД и в то же время достаточно удобные, для улучшения и изменения, отвечая требованиям поставленных задач. Утилиты для проверки структур таблиц и моделей данных обеспечивают высокую надёжность при разработке. Это избавляет разработчика от ошибок во время работы над новой ER-диаграммой или создания физической MySQL БД.

Программный продукт MySQL Workbench является бесплатным решением. Что даёт значительную экономию на программном обеспечении необходимом при разработке.

Основные преимущества программы:

- Функциональный и наглядный механизм установки связей между таблицами;
- Позволяет наглядно представить модель БД в графическом виде;
- Удобный редактор SQL запросов, позволяет моментально отправлять их серверу и получить ответ в табличном виде;

Изм.	Колоч.	Лист.	№ док	Подпись	Дата

- Reverse Engineering – восстановление структуры таблиц из уже существующих на сервере баз данных;
- Возможность редактирования данных в таблице в графическом режиме.

4.1.4 Система контроля версий Mercurial

Контроль версий представляет собой процесс управления несколькими версиями конкретной информации. Самым примитивным методом контроля версий можно считать каждое изменение файла пользователем, при котором происходит сохранение его под новым именем, в котором указан порядковый номер, каждый из которых выше, чем число предшествующей версии.

ПО Mercurial обладает особым набором свойств, позволяющим выбрать его в качестве наиболее подходящей системы управления ревизиями:

- Легковесный
- Прост в изучении и использовании
- Легко настраивается под конкретные нужды
- Превосходно масштабируется

Если пользователь уже обладает опытом в использовании систем контроля версий, то ему потребуется минимальное количество времени для того чтобы начать работать с Mercurial. Но даже в случае, если пользователь еще не работал с продуктом, процесс освоения будет очень прост. Mercurial предоставляет последовательную и единообразную систему функций и команд, что позволяет в процессе работы пользоваться небольшим набором общих правил вместо того, чтобы учить массу исключений.

В небольших проектах работа с Mercurial активизируется в считанные минуты. Создание новых изменений и веток, распространение изменений, операции со статусом и историей работает достаточно быстро.

Mercurial поддерживает не только маленькие проекты, его так же используют в проектах с сотнями и тысячами разработчиков, проектах, которые содержат сотни мегабайт исходного кода и десятки тысяч файлов.

Если вам не хватает базовых функций Mercurial, то их можно легко увеличить. Mercurial хорошо подходит для задач разработки скриптов, его понятная реализация на языке Python позволяет легко добавлять новый функционал в виде расширений. Существует большое количество полезных и достаточно популярных дополнений, охватывающих широкий спектр задач, от помощи в нахождении ошибок, до улучшения производительности.

Глава 5 Эргономика

5.1 Эргономика эффективное взаимодействия интерфейса «человек-компьютер»

Эргономика (от греч. *ergon* работа и *nomos* закон) – основным видом деятельности данной дисциплины является разработка и изучение эффективных систем, которые использует в своей деятельности конечный пользователь.

Эргономика – это наука способствующая приспособлению рабочих мест, должностных обязанностей, объектов и предметов труда, а также компьютерного обеспечения с целью улучшить эффективность труда работников, исходя из психических и физиологических особенностей человеческого организма. Эргономичность интерфейса является основным залогом продуктивной, эффективной и удобной работы пользователя в программе. Качественно проработанная эргономика интерфейса не должна негативно влиять на психологическое и физическое состояние пользователя, что положительно отразится на скорости и результативности деятельности человека.

Основным принципам разработки качественных интерфейсов принято считать:

- Гибкость – при разработки интерфейса необходимо предусмотреть, что с программой будут работать пользователи с различными уровнями подготовки;
- Не избыточность – от пользователя необходим ввода минимального объема информации;
- Естественность – взаимодействие с пользовательским интерфейсом не должно вызывать осложнений у пользователя;
- Непротиворечивость – взаимодействие с программой должно соответствовать привычным для работы принципам.

Изм.	Коллич.	Лист.	№ док	Подпись	Дата

5.2 «Usability» простота и удобство в использовании

Когда говорят о практичности, удобстве и простоте в использовании то, как правило, используют английское слово «Usability. Определение этого слова звучит как «степень успешности, эффективности и удовлетворения, с которыми продукт может быть использован определенными пользователями в определенном контексте использования для достижения определенных целей» (стандарт ISO DIS 9241-11).

Основная идея Usability заключается в определении оптимального интерфейса, который в удобной и простой форме предоставит пользователю всю необходимую функциональность, а также не будет ограничивать основные механизмы системы.

Определим основные требования по интерфейсной части для разрабатываемого продукта:

- максимальное взаимодействие с пользователем. Выводимая информация должна быть легко читаемой и понятной. А также необходимой, что позволило бы избавить пользователя от излишней информативности;
- минимальный объем информации, который пользователь должен запоминать для того чтобы использовать данную системы;
- учет навыков пользователя.

Основные требования были учтены при создании информационной системы, для поддержания работы системного администратора компьютерной сети.

Отличительной особенностью системы является реализация максимально удобного интерфейса, интуитивно понятного, для пользователя, при работе с которым, он не теряет время на выполнения определенных операций.

Изм.	Коллич.	Лист.	№ док	Подпись	Дата

ДП-230201.65-0803851 ПЗ

Лист

38

5.3 Интерфейс разработанной системы

В понимании слова «интерфейс» мы будем рассматривать некий механизм в определенной упрощенной форме, который позволяет человеку управлять объектом, не вдаваясь во внутренние зависимости. Интерфейс может скрывать от пользователя излишний функционал объекта, в котором для пользователя нет ничего необходимого.

Данный термин включает в себя не только программное обеспечение компьютера, но и аппаратное, дающее пользователю информацию и позволяет ему работать с компьютером.

Чаще всего под пользовательским интерфейсом воспринимают только внешний вид программного обеспечения. Но на самом деле он включает в себя множество аспектов, которые существенно влияют на взаимодействие системы и пользователя. Проектирование интерфейса основывается на знаниях, опыте и том, что пользователь ожидает от системы. Основными принципами для формирования интерфейса послужило следующее:

- использование минимального количества режимов с системой;
- обеспечение понятного интерфейса путем корректного изложения материала.

Интерфейс системы разработан с учетом особенностей сферы применения автоматизированной системы. Он ориентирован на пользователей знакомых с общими принципами работы Windows – приложений.

При запуске программы появляется окно для авторизации пользователя (рисунок 5.1)

После правильного введения данных необходимых для авторизаций, появляется сообщение об успешном подключении к базе данных (рисунок 5.2)

Изм.	Колич.	Лист.	№ док	Подпись	Дата

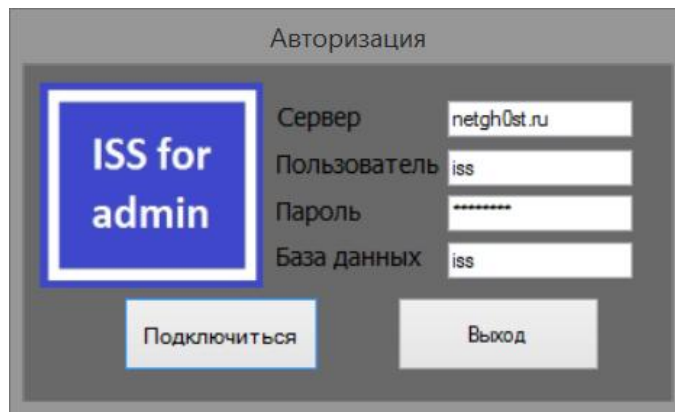


Рисунок 5.3 - Окно авторизации

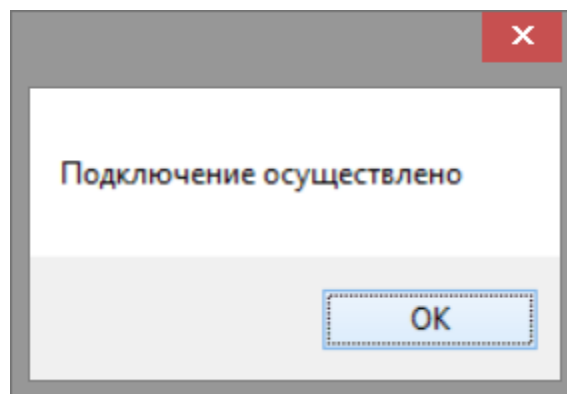


Рисунок 5.4 - Сообщение о подключении

Нажимая кнопку «ОК» происходит подтверждение пройденной авторизации и открывается основное окно с шестью объектами. (рисунок 5.3)

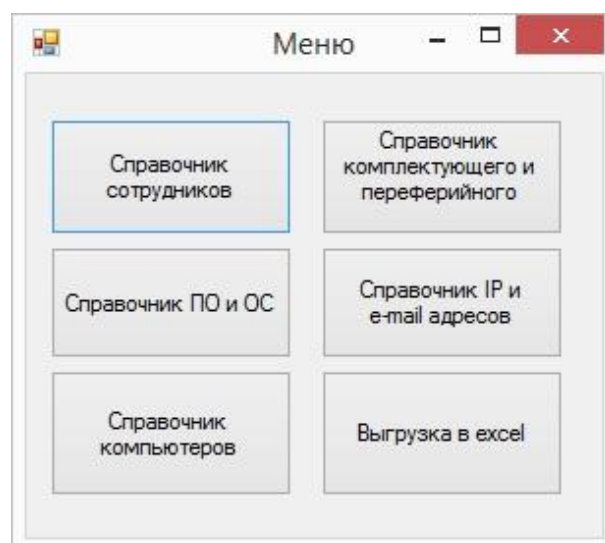


Рисунок 5.5 - Основное окно программы

Изм.	Колич.	Лист.	№ док	Подпись	Дата

Как видим на рисунке интерфейс основного окна очень простой. Использует системное оформление окна, не используются яркие цветовые решения, благодаря чему не происходит отвлечения внимания.

Рассмотрим окно справочника на примере справочника ip и e-mail адресов. Интерфейсы справочников очень схожи между собой и имеют чаще всего одинаковый набор компонентов и их расположение. (рисунок 5.4)

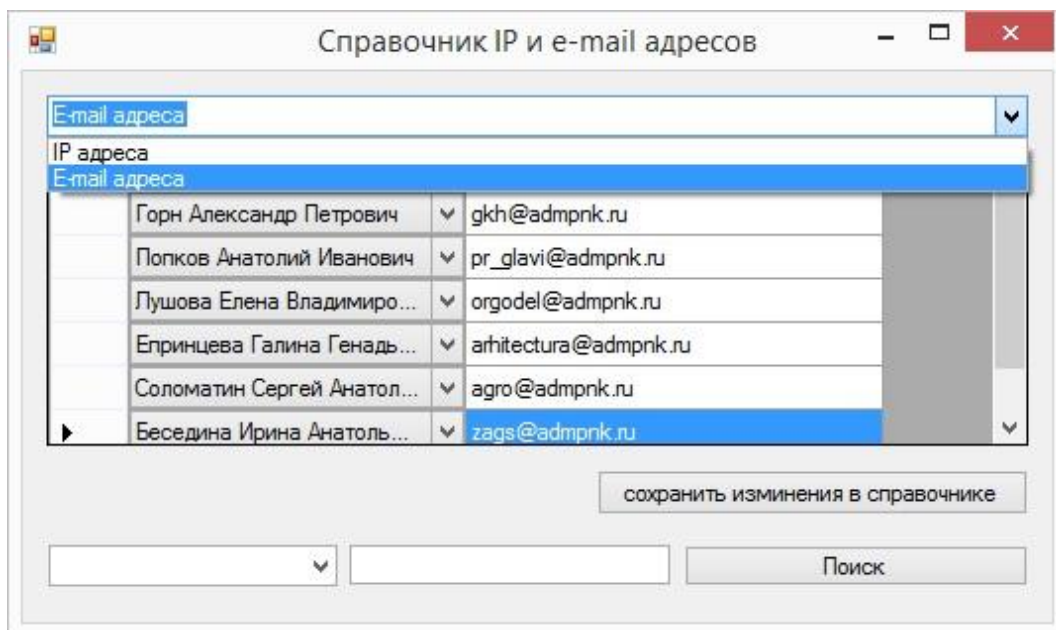


Рисунок 5.6 - Справочник IP и e-mail адресов

Оформление интерфейса разрабатывалось в максимально простом стиле. В силу этого пользователь не будет тратить время на изучение самой программы, а будет выполнять работу с минимальными затратами по времени. Эффективность взаимодействия системы и человека основана на графическом восприятии информации, которая достаточна, лаконична и информативна. Также на упрощенный графический интерфейс тратится наименьшее количество ресурсов вычислительной техники.

Изм.	Колич.	Лист.	№ док	Подпись	Дата

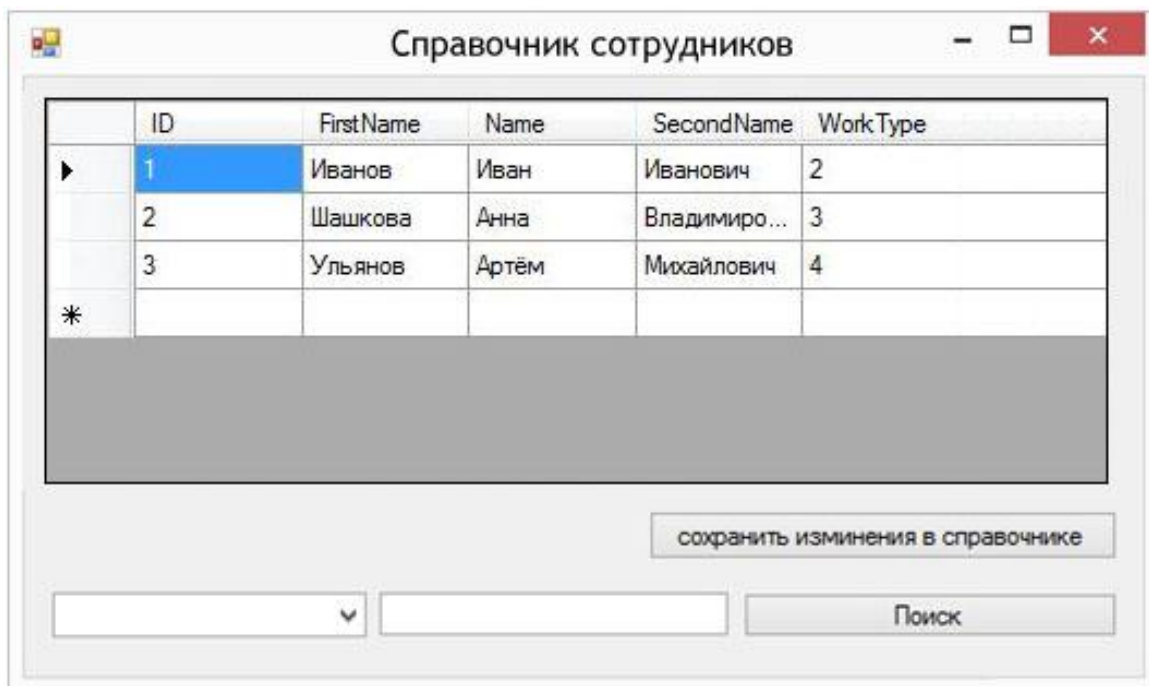


Рисунок 5.5 - Справочник сотрудников

Изм.	Колич.	Лист.	№ док	Подпись	Дата

Глава 6 Информационная безопасность

6.1 Перечень возможных угроз

Возможные угрозы - это совокупность факторов и условий, которые могут стать причиной нарушения информационной безопасности.

Угрозой принято считать потенциально возможное действие, событие, явление и процесс, которое может стать причиной нанесения ущерба чьим-либо интересам.

Угрозы информационной безопасности подразделяются по нескольким признакам.

По аспекту информационной безопасности, на который направлены угрозы:

- угроза доступности (осуществление действий затрудняющих или делающих невозможным доступ к ресурсам ИС);
- угрозы конфиденциальности (неинициализированный доступ к информации);
- угрозы целостности (неправомерное изменение данных).

По степени преднамеренности действий:

- преднамеренные (умышленные действия, например, диверсии и шпионаж);
- случайные (неумышленные действия, сбой в работе систем).

По расположению источника угроз:

- внешние (источник угроз находится вне системы);
- внутренние (источник угроз расположен внутри системы).

По размерам причиняемого ущерба:

- общие (нанесение значительного ущерба и нанесения ущерба объекту безопасности в целом);

Изм.	Коллич.	Лист.	№ док	Подпись	Дата

- частные (нанесение вреда отдельным свойствам элементов объекта безопасности);

- локальные (нанесение ущерба отдельным элементам объекта безопасности).

По степени воздействия на информационную систему:

- активные (содержание и сама структура системы подвергаются изменениям);

- пассивные (содержания и структура остаются неизменными).

Система имеет иметь клиент-серверную архитектуру. Клиентская часть системы разрабатывается под операционную систему семейства Windows с установленными библиотеками .NET не ниже версии 3.5. Серверная часть базируется на СУБД MySQL и не привязана к определенной системе. Поэтому она может находиться на системах Unix и Windows подобных.

Права доступа к тем или иным участкам данных регулируется самой системой СУБД. Что позволяет разграничивать доступ к базе данных для нескольких пользователей.

При рассмотрении вопросов проектируемой информационно справочной системы возможны следующие угрозы информационной безопасности:

- Сбой и вывод из строя оборудования
- Ошибки ПО
- Халатность при использовании системы
- Подмена ПО
- Повреждения каналов связи
- Несанкционированный доступ к информации

Изм.	Коллич.	Лист.	№ док	Подпись	Дата

6.2 Выбор методов и средств защиты информации

Так как в разработке прототипа информационно справочной системы используется СУБД MySQL, то и обеспечение безопасности информации осуществляется стандартными средствами данной СУБД.

Для защиты от угроз были выбраны следующие действия:

- Сбой и вывод из строя оборудования. Для решения данной проблемы предлагается установка бесперебойного питания, использование RAID массивов жестких дисков, а также резервное копирование информации.
- Ошибки ПО. Это ошибки реализации, проектирования и настройки Программного обеспечения. Для решения проблем на этапе проектирования нам необходимо согласовать техническое задание и поставленную задачу с заказчиком. Для решения проблем реализации проекта необходимо тестирование на оборудовании различной конфигурации.
- Несанкционированный доступ к информации. Для решения данной угрозы необходимо зарегистрировать двух пользователей одного с возможностью изменения и дополнения базы данных, другого пользователя только для чтения информации с базы данных. Пароли должны иметь длину не менее 8 символов, и использоваться должны оба регистра, а также числовые символы. Смена пароля не реже одного раза в месяц.
- Доступ к данным осуществляется через разрабатываемую информационно справочную систему посредством авторизации через имя пользователя и его пароль.

6.3 Создание архивов баз данных и их восстановление

Для быстрого и удобного создания резервных копий баз MySQL существует специальная программа `mysqldump`. Также `mysqldump` позволяет производить очень тонкие настройки для управления процессом создания

резервных копий БД или отдельных таблиц. Данную программу можно назвать основным инструментом, если есть необходимость в создании резервных копий баз данных MySQL. Программа `mysqldump` производит вывод результатов, как на `STDIN`, так и в какой-либо файл. Вывод `mysqldump` - это не какие-то абстрактные и не читаемые двоичные данные, а осмысленный текст сценария. Для более тонкой настройки работы `mysqldump` предусмотрен обширный набор ключей.

Более тонкие настройки `mysqldump`:

- `databases` позволяет сделать так, что `mysqldump` включает в сценарий восстановления команды `create database dbname` и `use dbname`. Это дает возможность пользователю создавать новые рабочие базы. Без использования `databases` пользователю явно указывать, куда нужно помещать восстанавливаемые данные;

- `all-databases` дает возможность сделать копии всех БД, которые существуют на используемом MySQL-сервере. Если нужно сделать резервные копии определенных баз, то нужно указывать их через пробел при вызове `mysqldump` из командной строки;

- `add-drop-table` - ключ, который добавляет в конечный сценарий команду `drop table` перед созданием таблиц. Это позволяет избежать определенных ошибок при восстановлении базы из резервной копии. Следует учитывать то, что таблицы, находящиеся в рабочей копии, перед восстановлением из резервной копии будут удалены из основной базы и пересозданы из резервной копии;

- `result-file=...` - этот ключ можно использовать для перенаправления вывода в файл.

`Mysqldump` имеет еще некоторое количество полезных возможностей, которые Вы можете применять по обстоятельствам.

Для защиты данных и программного обеспечения от внедрения вредоносных программ и вирусных угроз необходимо установить антивирусное ПО, а также регулярно проводить обновления их баз.

В данном разделе были рассмотрены некоторые виды угроз информационной безопасности, и описаны методы их решения.

						<i>ДП-230201.65-0803851 ПЗ</i>	<i>Лист</i>
<i>Изм.</i>	<i>Коллич.</i>	<i>Лист.</i>	<i>№ док</i>	<i>Подпись</i>	<i>Дата</i>		47

ЗАКЛЮЧЕНИЕ

При написании дипломной работы были изучены механизмы работы с СУБД MySQL, а так изучены технологии .NET на основе языка C# входящего в состав Visual Studio 2015 Community Edition.

Произведены работы по моделированию функциональных требований к проектируемой информационно справочной системе, работы по разработке логической модели данных в виде диаграммы классов, а также описание структуры входных и выходных данных.

Спрограммирован прототип информационной системы.

При анализе эргономичности информационной системы были определены основные требования, спроектирован и реализован пользовательский простой интерфейс.

Проведен анализ информационной безопасности и рисков, связанных с ними, так же определен перечень возможных угроз, методы и средства защиты от них. Так же был рассмотрен стандартный способ СУБД MySQL по резервному копированию и восстановлению базы данных.

В ходе дипломной работы, разработана информационная система для поддержания деятельности системного администратора компьютерной сети. Информационная система не имеет ограничений и может быть применена системным администратором любой организацией для осуществления своей деятельности. Возможность гибкой настройки позволяет добиться наибольшей эффективности при использовании продукта.

Благодаря удобному пользовательскому интерфейсу информационной системы для поддержания деятельности системного администратора компьютерной сети, пользователь не будут уставать от однообразной работы, а также обучение работы в программе пройдет намного быстрее, так как интерфейс понятен и прост в обращении.

Изм.	Коллич.	Лист.	№ док	Подпись	Дата

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

- 1 Боуман, Дж. Практическое руководство по SQL. : учебное пособие / Дж. Боуман, С. Эмерсон, М. Дарновски. – Киев : Диалектика, 1997. - 200 с.
- 2 Бурков, П. В. Введение в системы баз данных : учебное пособие / П. В. Бурков, В. Ю. Петров. – Санкт-Петербург : СПбГУ ИТМО, 2010. – 129 с.
- 3 Буч, Г. Язык UML: руководство пользователя : учебное пособие / Г. Буч, Дж. Рамбо, А. Джекобсон ; под общ. ред. А. А. Симкина. – Санкт-Петербург : Питер, 2004. – 432 с.
- 4 Гайдамакин, Н. А. Автоматизированные системы, базы и банки данных. Вводный курс : учебное пособие / Н. А. Гайдамакин. – Москва : Гелиос АРВ, 2002. – 368 с.
- 5 Глушкова, С. В. Базы данных : учебник / С. В. Глушкова, Д. В. Ломотько. – Харьков : Фолио, 2002. – 504 с.
- 6 Яркова, С. А. Дипломная работа. Структура. Требования к содержанию и оформлению: метод. рекомендации / С. А. Яркова, Г. А. Доррер, Л. В. Ефимова – Красноярск : СибГТУ, 2010. – 40 с.
- 7 Избачков, Ю. С. Информационные технологии : учебное пособие / Ю. С. Избачков. – Санкт-Петербург : Питер, 2009. – 272 с.
- 8 Ларман, К. Применение UML и шаблонов проектирования : учебник / К. Ларман – Москва : Гелиос АРВ, 2004. – 176 с
- 9 Курлаев, С. А. Распределённые информационные системы и базы данных : учебное пособие / С. А. Курлаев, И. А. Цильковский. – Новосибирск : НГТУ, 2012. – 92 с.
- 10 Лопатников, Л. И. Экономико-математический словарь: Словарь современной экономической науки : словарь / Л. И. Лопатников. - 5-е изд., перераб. и доп. – Москва : Дело, 2003. - 520 с.
- 11 Маклаков, С. В. BPWin и ERWin. CASE-средства разработки информационных систем : учебное пособие / С. В. Маклаков. – Москва: ДИАЛОГ-МИФИ, 2000. – 300с.

						<i>ДП-230201.65-0803851 ПЗ</i>	Лист
							49
<i>Изм.</i>	<i>Коллич.</i>	<i>Лист.</i>	<i>№ док</i>	<i>Подпись</i>	<i>Дата</i>		

12 Роб, П. Системы баз данных: проектирование, реализация и управление. / П. Роб, К. Корнел. – 5-е изд., перераб. и доп. – Санкт-Петербург : БХВ-Пертебург, 2004. – 1040 с.

13 Сычев, Ю. Н. Информационная безопасность, учебно-практическое пособие : учебное пособие / Ю. Н. Сычев. – Москва : Изд. центр ЕАОИ, 2007. – 300 с.

14 Фаулер, М. UML. Основы : учебное пособие / М. Фаулер. – Санкт-Петербург : Символ-Плюс, 2004. –192 с.

15 Федотова, Д. Э. CASE – технологии. Практикум: пособие для вузов : учебное пособие / Д. Э. Федотова, Ю. Д. Семенов, К. Н. Чижик. – Москва : Горячая линия-Телеком, 2005. – 160 с.

16 Шилд, Г. Полный справочник по С#. / Г. Шилд. – Москва : Издательский дом «Вильямс», 2004. – 752 с.

17 Хомоненко, А. Д. Базы данных: учебник для высших учебных заведений / под ред. проф. А. Д. Хомоненко. – Санкт-Петербург : КОРОНА принт, 2004. – 736с.