**Правительство Российской Федерации**

**Федеральное государственное автономное образовательное**

**учреждение высшего профессионального образования**

**"Национальный исследовательский университет**

**"Высшая школа экономики"**

***Отделение программной инженерии***

***Кафедра Управления разработкой программного обеспечения***

УТВЕРЖДАЮ

Зав. кафедрой УРПО

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ С.М. Авдошин

«\_\_» \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ 20\_\_ г.

**ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА**

***по направлению 231000.62 Программная инженерия***

***подготовки бакалавра***

На тему: **“Программа учета материальных ресурсов территориально-распределенного предприятия”**

Студента группы № 471ПИ \_\_\_\_\_\_\_\_\_\_\_ Туманова Романа Игоревича

Подпись (Ф.И.О.)

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(Дата)

Научный руководитель:

к.т.н., доцент \_\_\_\_\_\_\_\_\_\_\_ Брейман Александр Давидович

(должность, звание) Подпись ( Ф.И.О.)

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(Дата)

Москва, 2014 г.

# Реферат

Количество страниц: 37

Количество глав: 3

Количество иллюстраций: 22

Количество приложений: 4

Количество использованных источников: 9

В данной выпускной квалификационной работе рассматривается разработка программы для учета материальных ресурсов небольшого территориально – распределенного предприятия. Специфика данного предприятия предполагает небольшое количество пользователей системы, которые должны обладать базовыми навыками работы с компьютером, что подразумевает интуитивно понятный интерфейс и простоту использования программы.

Целью данной работы является: создание программы, которая позволяет вести учет ресурсов территориально-распределенного предприятия с небольшим парком рабочих станций, используя самостоятельную базу данных. Данная работа имеет практическую значимость в связи с тем, что предполагается ее использование на реальном предприятии, автоматизация учетной деятельности которого должна принести положительные с точки зрения эффективности работы результаты. Результатом работы является реализация такой программы, которая полностью учитывает требования данного предприятия и не несет в себе избыточного функционала.

Во вводной части приведены актуальность работы, цели и задачи. В первой части данной работы приведен анализ существующих решений для данной проблемы, выбор средств для разработки. Вторая часть содержит сведения об архитектуре программы и о структуре базы данных. В третьей части рассматриваются модули программы, основные классы, работа пользователей в программе в зависимости от разных прав доступа. В заключении приведены результаты данной работы, область дальнейшего применения.

Ключевые слова: территориально – распределенное предприятие, ERP (Enterprise Resource Planning), склад, материальные ресурсы, база данных, автоматизация, отчетность.

Содержание

**Введение**4

Актуальность работы4

Цель5

Задачи5

Программные требования5

**Глава 1. Существующие решения и выбор технологии**7

1.1 «1С: Управление производственным предприятием 8»7

1.2 «Microsoft Navision Axapta»8

1.3 Выводы по двум системам 8

1.4 Технология 9

1.5 Требования 10

**Глава 2. Архитектура программы** 13

2.1 Архитектура программы 13

2.2 Структура БД 15

**Глава 3. Реализация** 19

3.1 Пользовательский интерфейс 19

3.2 Модули программы 22

3.3 Описание классов программы 28

**Заключение** 36

**Список использованных источников**37

**Приложение А. Техническое задание**38

**Приложение Б. Руководство оператора**51

**Приложение В. Программа и методика испытаний**82

**Приложение Г. Текст программы**97

# Введение

## Актуальность работы

Эффективность любого современного предприятия напрямую зависит от наличия автоматизации процессов этого предприятия. Речь, прежде всего, идет о ERP – системах [5], которые, начиная с 1990-х, внедряются не только в промышленных предприятиях, но и в других предприятиях независимо от страны и рода деятельности. Внедрение подобных систем для автоматизации учета ресурсов предприятия дает положительный эффект не только в крупных организациях, но так же в малом и среднем секторе, и является необходимым аспектом успешного, продуктивного и прибыльного функционирования предприятия. Есть несколько важных аспектов, которые определяют уровень автоматизации.

Прежде всего, очень важна сфера деятельности компании. Для стандартного предприятия есть много готовых решений (off-the-shelf software) на рынке программного обеспечения. Однако многие фирмы имеют ряд особенностей в процессе производства и в учете и контроле ресурсов, которые не позволяют использовать готовые решения. Такие компании вынуждены настраивать или изменять готовые программные системы. Подобные улучшения и настройки иногда обходятся дороже, чем стоимость самой системы.

Во-вторых, есть два важных аспекта компании: размер компании и годовой оборот дохода. Как правило, эти значения находятся в прямой зависимости, но исключения из этого правила встречаются. В большинстве случаев, автоматизация деятельности крупной компании, с несколькими филиалами, расположенными в разных частях города, требует сложную и развитую систему автоматизации. Она включает в себя развитую сетевую инфраструктуру, постоянно обновляемый парк рабочих станций и серверов с эффективным персоналом сетевых и системных администраторов.

## Цель

Создание программы, которая позволяет вести учет ресурсов реального территориально-распределенного предприятия [8] с небольшим парком рабочих станций(5-30), используя самостоятельную базу данных. Данная программа должна учитывать все особенности данного предприятия, иметь дружественный интерфейс, и возможность для быстрой модификации.

## Задачи

1. Проектирование базы данных предприятия;
2. Разработка архитектуры программы;
3. Разработка интерфейса программы;
4. Создание прототипа программы, работа с базой данных;
5. Реализация аутентификации пользователей, создание групп пользователей с разными правами;
6. Разработка модулей программы;
7. Тестирование;

## Программные требования:

1. Проектирование базы данных является необходимой и базовой частью данной работы, т.к. все данные предприятия хранятся в базе;

2. В программе должны учитываться различные права доступа, например, только администратор имеет возможность добавлять, редактировать и удалять сотрудников из базы;

3. Программа должна осуществлять подключение к БД предприятия;

4. Программа должна отслеживать корректность вводимых пользователем данных, и сообщать пользователю об ошибках;

5. Приложение должно обладать интуитивным графическим интерфейсом;

6. Клиентская часть:

1) Ввод, изменение и удаление данных о работниках предприятия, продукции, комплектах, производимых географически распределенными командами работников.

2) Создание отчетов и другой документации предприятия;

3) Получение информации о денежных выплатах работникам в заданном диапазоне времени;

4) Учет и мониторинг состояния склада;

5) Автоматическая генерация документов указанного типа, отражающих движение материалов и комплектов;

6) Проверка наличия материалов на складе, необходимых для производства данного продукта, сборок и / или наборов с заданными параметрами;

# Глава 1. Существующие решения и выбор технологии

В этой главе будут рассмотрены существующие решения, позволяющие проводить учет ресурсов предприятия. Существует множество аналогов, в данной главе будут рассмотрены 2 приложения, имеющие богатый, но избыточный для данного предприятия функционал.

## 1.1 «1С: Управление производственным предприятием 8»

Прежде всего, будет рассмотрена система «1С: Управление производственным предприятием 8» [6]. «1С: Управление производственным предприятием 8» представляет собой интегрированное решение, которое охватывает основные области управления и учета в производственной компании. Применяемое решение создает единое информационное пространство для отображения финансово-хозяйственной деятельности предприятия, охватывая основные бизнес-процессы. В то же время, данная система помогает четко дифференцировать доступ к хранимым данным, а также возможность определенных действий в зависимости от состояния рабочих.

«1С: Управление производственным предприятием 8» обеспечивает:

• Для руководства предприятия - широкий спектр анализа, планирования и гибкого управления ресурсами для повышения своей конкурентоспособности;

• Для руководителей отделов - инструменты для повышения эффективности ежедневной работы в своих областях;

• Для сотрудников компании - инструменты для автоматического учета в полном соответствии с требованиями законодательства и корпоративными стандартами предприятия.

Данный функционал в рамках данной конкретной задачи избыточен, т.к. предприятие для внедрения небольшое, следовательно, ему не требуется большое разнообразие функционала. Избыточность функционала приведет так же к тому, что малочисленный штат сотрудников придется обучать работе с системой. Компанией 1С проводятся множество курсов по работе с системой и это однозначно плюс для такого серьезного и многофункционального продукта, однако такие курсы это отдельный пункт в графе затрат на систему управления материальными ресурсами предприятия, что в совокупности со значительной стоимостью одной копии для 1 машины составляет непропорциональные требуемому функционалу затраты.

Кроме того, ожидается, что наибольший эффект от внедрения этого решения будет на предприятиях с персоналом от нескольких десятков до нескольких тысяч человек, которые имеют десятки или сотни рабочих станций. Однако предприятие, для которого разрабатывается данный продукт, имеет около 20 рабочих мест.

## 1.2 «Microsoft Dynamics AX»

Это интегрированная система управления класса ERP II (Enterprise Resource & Planning) [5], объединяет различные отделы и услуги предприятия в едином информационном пространстве. Это гарантирует качество управленческой информации и позволяет менеджерам принимать обоснованные управленческие решения.

Microsoft Dynamics AX [9] имеет следующие модули:

• Финансовый менеджмент;

• Логистика и торговля;

• Производство;

• Стратегическое управление;

Axapta предлагает интуитивно понятный интерфейс: дизайн меню и панели инструментов, синтаксис поисковой системы и терминологии заимствованные из пакета Microsoft Office. В Microsoft Dynamics AX система обеспечивает простые и удобные инструменты публикации данных на корпоративном веб-сайте. Эти инструменты основаны на автоматической генерации веб-страниц в интернет-сервер с указанным интервалом.

## 1.3 Выводы по двум системам

Подводя итог, следует отметить, что рассмотренные системы относятся к продуктам примерно одного и того же класса, а функциональность любого из них является избыточной в рассмотрении данной задачи автоматизации. Кроме того, внедрение продуктов такого уровня является очень дорогостоящим для малых предприятий. Например, цена на Microsoft Dynamics AX на станции колеблется от € 800 до € 1750. Такие расходы получаются не из-за высокой начальной стоимости продукта, а из-за расходов на поддержку и конфигурацию системы под требования заказчика.

В свете всего вышесказанного, было принято решение о разработке программы для данного предприятия.

## 1.4 Технология

Язык программирования Java [1] был выбран для реализации. Прежде всего, этот язык мультиплатформенный, что позволяет использовать эту систему на всех машинах, независимо от их операционной системы. Несмотря на то, что компьютеры на предприятии работают под операционной системой Windows, удаленные пользователи могут войти в систему на компьютере под управлением другой операционной системы. Эта функция на самом очень важна, потому что, если кто-то из сотрудников, занимающих руководящие должности хочет войти в систему с персонального компьютера в деловой поездке, он или она может сделать это в любое время. Без мультиплатформенности, возникнет множество неожиданных задержек в принятии важных решений по управлению предприятием. В-вторых, JAVA является объектно-ориентированным языком, который позволяет создавать модульные программы, чей исходный код может быть использован повторно. Для хранения данных в базе данных была выбрана MySQL[7]. База данных MySQL имеет высокую производительность вместе с высокой надежностью. Есть несколько важных особенностей, которые имеют решающее значение для этого проекта: многопоточность, поддержка нескольких одновременных запросов, гибкая система привилегий и паролей, хорошая поддержка со стороны поставщиков услуг хостинга. Среда IDE NetBeans [2] была выбрана в качестве среды разработки. Это среда разработки имеет интуитивно понятный интерфейс. Кроме того, она идеально подходит для Java. Для взаимодействия с БД и для решения задач объектно-реляционного отображения была использована библиотека Hibernate[3], которая представляет собой свободное программное обеспечение с открытым исходным кодом.

## 1.5 Требования

## Требования к функциональным характеристикам

Программа должна обеспечивать возможность выполнения перечисленных функций.

1. Доступ к данным, основанный на правах доступа пользователей, пользовательские роли делятся на:

* Пользователь
* Товаровед
* Бухгалтер
* Администратор

В зависимости от роли каждому пользователю программы доступны определенные функции.

1. Загрузка данных в приложение из базы предприятия в зависимости от прав доступа:

В зависимости от роли, каждый пользователь может просматривать определенную информацию, полученную из БД предприятия.

1. Запись данных в базу предприятия. Запись в БД предприятия могут вести только следующие группы пользователей:

* Товаровед
* Бухгалтер
* Администратор

Пользователь не имеет прав по изменению данных в базе, ему доступно только чтение.

1. Удаление записей в базе данных предприятия: удаление записей основано на правах доступа описанных выше. Пользователь не может удалять данные из БД предприятия.
2. Расчет выплат сотрудникам: доступен Бухгалтеру и Администратору программы, позволяет рассчитывать выплаты рабочим за выбранный промежуток времени.
3. Генерация документов, доступна Товароведу и Администратору программы.

## Требования к интерфейсу

В левой части окна программы должно быть меню, с помощью которого осуществляется переход между модулями программы.

## Требования к надежности

Программа осуществляет проверку вводимых пользователем данных на корректность.

## Условия эксплуатации

Программа не требует специального обслуживания. Пользователь должен

посмотреть раздел меню «Помощь» для ознакомления с основными возможностями программы. Требуемая квалификация пользователя - оператор.

## Требования к составу и параметрам технических средств

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Необходимый процессор | Рекомендуемый процессор | Необходимое ОЗУ | Рекомендуемое ОЗУ | Прочие требования |
| 32-разрядный (x86) или 64-разрядный (x64) процессор с тактовой частотой 1 гигагерц (ГГц) | 32-разрядный (x86) или 64-разрядный (x64) процессор с тактовой частотой 1,6 гигагерц (ГГц) или выше | 1 гигабайт (ГБ) (для 32-разрядной системы) или 2 ГБ (для 64-разрядной системы) оперативной памяти (ОЗУ) | 2 гигабайт (ГБ) (для 32-разрядной системы) или 4 ГБ (для 64-разрядной системы) оперативной памяти (ОЗУ) | графическое устройство DirectX 9 с драйвером WDDM версии 1.0 или выше |

\*Минимум, требуемый операционной системой Windows7.

* Клавиатура и мышь или совместимое указывающее устройство
* Привод CD-RW/DVD-RW

## Требования к программной и информационной совместимости

Программные средства:

-программа реализована на языке Java;

-среда программирования- NetBeans IDE 8.0;

-при разработке базы данных использована СУБД MySQL.

## Требования к маркировке и упаковке

## Программа предоставляется пользователю на Flash-устройстве с установщиком.

# Глава 2. Реализация приложения

В этой главе будут рассмотрены основы реализации приложения, которые включают в себя, схемы базы данных и архитектуру программы.

## 

## 2.1 Архитектура программы

Система представляет собой клиент-серверное приложение, которое состоит из двух частей:

1. Клиентского программного модуля написанного с использованием Java и развернутого на рабочих местах сотрудников;
2. Базы данных, созданной под управлением MySQL.

Приложение предполагает одновременное использование несколькими сотрудниками, изначально существовало 3 концепции для будущей системы:

Во-первых, это может быть приложение для одного пользователя. Этот вариант является наиболее простым для достижения цели этой работы. База данных и клиентское приложение, которое использует базу данных, расположены на одном физическом компьютере. Следовательно, нет никакой необходимости в обеспечении надежной связи между клиентом и базой данных. Данный вариант реализации был признан не эффективным для использования: для одновременной интенсивной работы в качестве клиентского приложения и базы данных, рабочие станции должны иметь относительно большие вычислительные ресурсы; все преимущества выделенной базы данных в этом случае аннулируется; приложение должно использоваться сотрудниками различных ведомств, что в данном случае невозможно;

       Во-вторых, это может быть система, которая предназначена только для использования в локальной сети предприятия. Выделенный сервер установлен для СУБД (система управления базами данных), и в этом случае необходимо позаботиться о надежной и быстрой связи клиентов на рабочих местах с сервером. С точки зрения использования функциональных возможностей системы, такая схема является гораздо более эффективной. Сотрудники различных подразделений могут использовать систему для конкретных потребностей каждого подразделения. Менеджеры и руководители могут использовать информацию и получать отчеты о бракованной продукции; кладовщики могут получить информацию о материалах и продукции на складе и на передвижение материалов и изделий; персонала и учет персонала и смогут получить данные о заработной плате работника.

        Наконец, система, которая будет использоваться в локальной сети и через Интернет. Этот вариант похож на предыдущий, однако есть и удаленные пользователи, добавленные в этом типе системы. Они могут получить доступ через Интернет с использованием VPN (Virtual Private Network) или модем. Была выбран третий тип системы, т.к. он имеет множество преимуществ перед первыми. Этот тип модели был выбран из-за больших возможностей по сравнению с двумя предыдущими. Такая схема позволяет использовать все возможности, как локальной сети, так и удаленного доступа (Рис. 1). Как видно из схемы, есть два типа пользователей: локальные и удаленные клиенты.

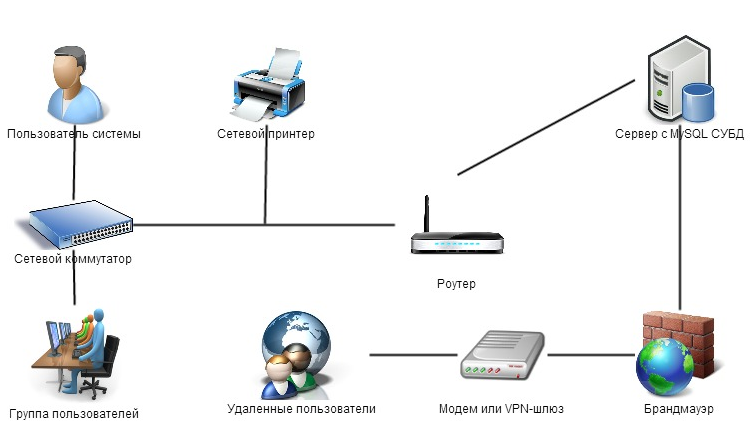


Рисунок 1. Архитектура программы

## Локальные возможности программы используются в главном офисе предприятия, они соединены в общую сеть. Далее к общей сети подключен сетевой принтер, который позволяет печатать созданные в программе документы: отчеты, инвойсы, накладные. Далее вся эта сеть соединена с сервером, на котором расположена база данных предприятия. Удаленные пользователи, это например руководители компании, которым в деловой поездке необходимо проверить склад предприятия, состояние выполненной работы и другие показатели. К удаленным пользователям относятся сотрудники, которые территориально находятся вне главного офиса, они посредством интернет соединения получают доступ к базе, на основании прав доступа, которые они получили от администратора.

## 2.2 Структура БД

### 2.2.1 Основные сущности

## Для данного приложения, была разработана база данных, которая отражает деятельность предприятия.

## C:\Users\Asus\Desktop\схемалюдей.png

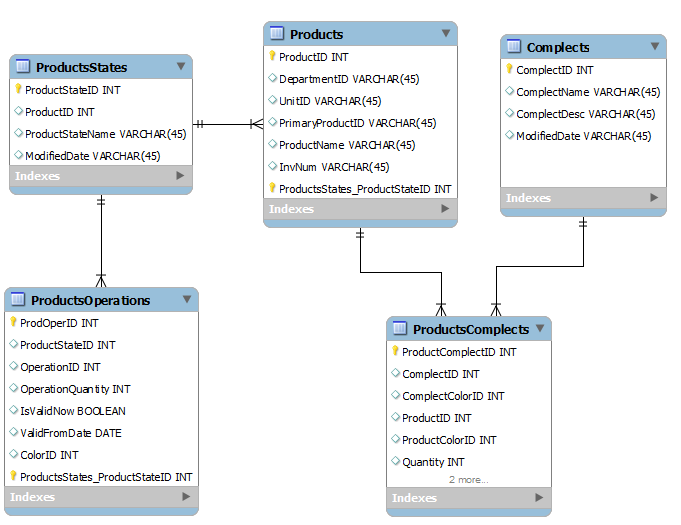
## Рисунок 2. Сотрудники

Созданная база данных состоит из 3 основным частей и 4 части, которая представляет собой набор вспомогательных таблиц.

Первая часть отражает информацию о людях, которые вовлечены в деятельность предприятия, это его сотрудники (Рис. 2). Сотрудники делятся на 3 типа: рабочие (самый важный тип), водители и пользователи системы. Как видно из схемы каждый из этих типов связан с надтипом «Контакты», в котором храниться все общая информация, присущая всем сотрудникам: Имя, Фамилия и т.д.

В подтипе «Пользователь» хранится логин и пароль пользователя для входа в систему. В подтипе «Водитель» не требуются дополнительные данные, поэтому он состоит поля первичного и внешнего ключа, для связи с таблицей «Контакты». Подтип «Рабочий» содержит даты его приема на работу, дату увольнения, ссылку на его текущую должность и дату изменения данных о рабочем.

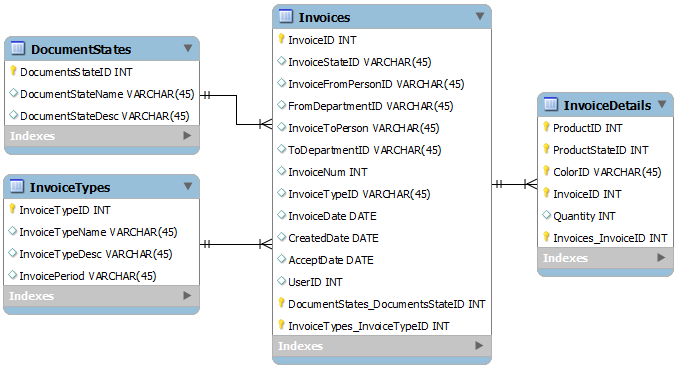
Следующая сущность это «Продукты» (Рис. 3).



## Рисунок 3. Продукты

Основной таблицей в данной схеме является таблица Products. Она содержит ссылку на департамент происхождения продукта, единицы измерения продукта, ссылку на первичный для этого продукта материал, наименование и внутренний инвентарный код. У каждого продукта есть свой набор состояний, которые представлены в таблице ProductStates. Для каждого состояния продукта существует набор операций, которые необходимы для перехода продукта в это состояние. Так же присутствует таблица Complects, она содержит комплекты готовых изделий, т.к. каждое изделие состоит из нескольких частей из таблицы Products. Связь между продуктами и комплектами осуществляется посредством таблицы ProductsComplects, в которой представлены ссылки на комплект и продукты.

Последним из основных блоков является блок «Документы».



## Рисунок 4. Документы

Основная таблица сущность «Документы» таблица Invioces. Атрибутами этой сущности являются ссылка на текущее состояние документа (таблица DocumentStates), тип документа (таблица InvoiceTypes), данные о документе (таблица InvoiceDetails).

### 2.2.2 Вспомогательные сущности

Помимо перечисленных выше трех основных сущностей в базе данных присутствует ряд более мелких вспомогательных. Проектируемая база состоит из 30 таблиц, ранее были показаны самые основные. Вспомогательные таблицы необходимы для описания товаров и документов, это справочники. Они описывают цвета продуктов, департаменты, виды документов, операции, производимые над продуктами и другие параметры базы:

* Colors (цвета материалов и продуктов),
* DefectsCauses (причины дефектов продукции),
* DocumentsStates (состояния документов),
* InvoiceTypes (типы документов),
* EquipmentTypes (типы оборудования),
* Position (должности работников),
* Units (единицы измерения продуктов, например шт.),
* WorkerStates (текущие состояния работников),
* Departments (департаменты, с которыми связаны продукты),
* Operations (операции, которые определены для работников),
* Requests (заявки на продукцию),
* RequestsDetails (связана с таблицей Requests в общую сущность «Заявки»),
* Stores и StoreDetails представляют сущность «Склад».

# Глава 3. Реализация

## 3.1 Пользовательский интерфейс

## При создании интерфейса важно чтобы он был интуитивно понятным пользователем, другими словами реализовывал принцип «делай то, что я имею в виду» или DWIM (Do What I Mean). Пользователь должен интуитивно понимать к чему приведет то или иное его действие, например, если пользователь хочет сгенерировать отчет\накладную, он должен четко понимать в каком именно месте программы должна находится данная функция и должен иметь возможность быстро найти эту функцию и использовать. В данной работе пользовательский интерфейс был разработан отталкиваясь от этого основного принципа и от следующих важных принципов:

* Интуитивность и простота.
* Визуальная привлекательность и удобство.
* Лаконичность и неперегруженность элементами.
* Реакция на события и действия пользователя.
* Информативность.

## Пользовательский интерфейс представляет собой окно, в левой части которого находится меню приложения, оно программно и визуально отделено от основного окна приложения, которое расположено в правой части окна (Рисунок 3.1.1 Пользовательский интерфейс). Меню представляет собой доступ к модулям программы, которые отвечают за определенную часть функциональности программы. При выборе модуля в меню в правой части окна появляется соответствующий модуль программы.

## Данная концепция была выбрана и разработана для того, чтобы минимизировать количество открытых окон данной программы. Такой интерфейс не перегружает визуально экран пользователя, и программа находится в одном окне, что позволяет быстро и эффективно находить все необходимые пользователю функции. Каждый модуль представляет собой окно, которое содержит в себе вкладки, позволяющие легко ориентироваться в каждом модуле делающие программу более компактной и лаконичной.

## Каждая кнопка снабжена пиктограммой, что увеличивает скорость нахождения нужного модуля и нужной функции в программе. В интерфейсе было использовано множество разнообразных элементов Swing, включая кнопки, выпадающие списки, таблицы, объединяющие панели, разделители панелей и компонент JDateChooser, который позволяет легко и быстро выбрать необходимую дату. В целом все сделано для того, чтобы пользователь, впервые столкнувшийся с программой мог быстро в ней разобраться.

## Скриншот (26.05.2014 00-51-40).jpg

**Рисунок 3.1.1 Пользовательский интерфейс**

## При запуске программы пользователя встречает форма аутентификации (Рисунок 3.1.2 Форма аутентификации). В данном окне пользователь вводит свой логин и пароль для входа в программу, которые были назначены администратором.

## Скриншот (21.05.2014 11-27-38).jpg

**Рисунок 3.1.2 Форма аутентификации**

## Программа осуществляет вывод сообщений пользователю (Рисунок 3.1.3 Сообщение об ошибке) при неверных действиях, и выделяет области, в которых пользователем были допущены ошибки ввода информации (Рисунок 3.1.4 Выделение поля ввода).

## Скриншот (22.05.2014 09-57-51).jpg

**Рисунок 3.1.3 Сообщение об ошибке**

## Скриншот (22.05.2014 10-01-46).jpg

**Рисунок 3.1.4 Выделение поля ввода**

## Отчеты в программе генерируются виде PDF-файлов, которые автоматически открываются программой после генерации при помощи программ для чтения PDF-файлов установленных на компьютере пользователя (Рисунок 3.1.5 Сгенерированная накладная).

## Скриншот (21.05.2014 13-22-37).jpg

**Рисунок 3.1.5 Сгенерированная накладная**

## 3.2 Модули программы

## В результате проектирования архитектуры программы и анализа требований было выделено 7 ключевых модулей программы:

## Сотрудники

## Работники

## Документы

## Расчеты

## Администрирование

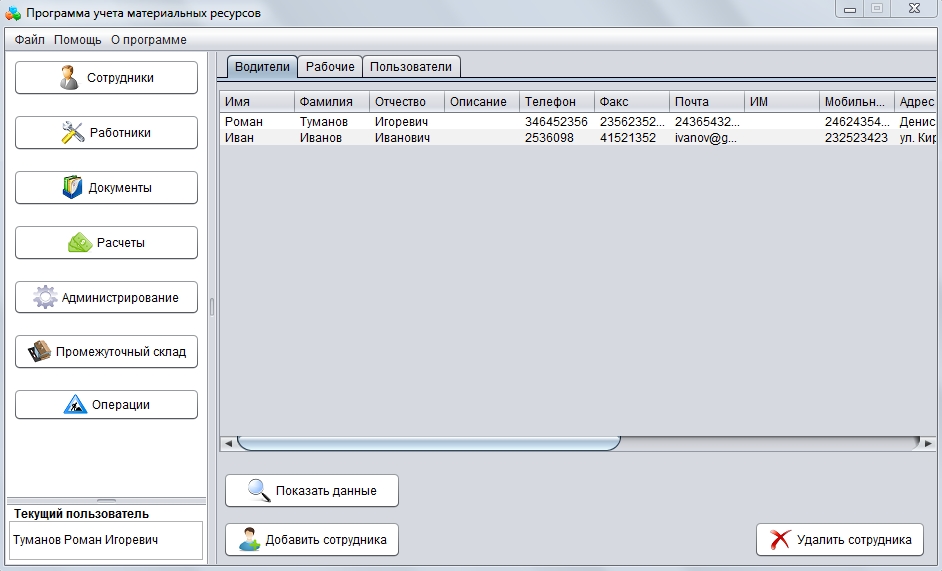
## Промежуточный склад

## Операции

## Каждый из этих модулей несет свой функционал, однако они связаны между собой, т.к. используют одну базу данных, таким образом изменение данных в 1 модуле повлечет за собой изменение во всех остальных. Рассмотрим каждый модуль подробнее.

### 3.2.1 Модуль «Сотрудники»

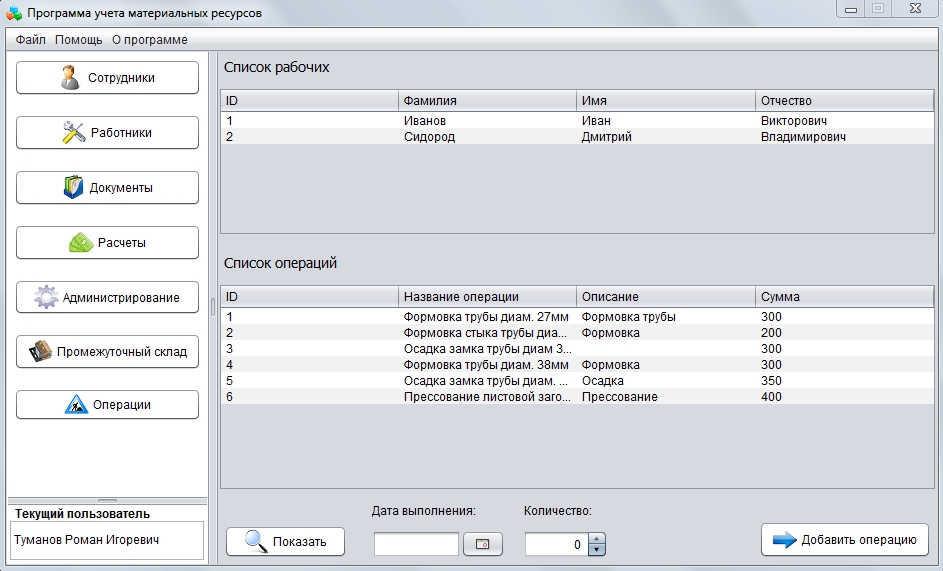
Данный модуль позволяет просматривать, создавать и удалять записи о сотрудниках предприятия используя базу данных предприятия (Рисунок 3.1.1.1 Модуль «Сотрудники»). Данный модуль имеет 3 вкладки: Водители, Рабочие, Пользователи. Удаление и добавление новых сотрудников в базу доступно только администратору.



**Рисунок 3.2.1.1 Модуль «Сотрудники»**

### 3.2.2 Модуль «Работники»

Данный модуль позволяет вести записи о проделанной сотрудниками по сборке продуктов работе (Рисунок 3.2.2.1 Модуль «Сотрудники»). Данный модуль имеет предельно простое управление и в случае неправильных действий, пользователь получит сообщение об ошибке, и о причине ее возникновения.

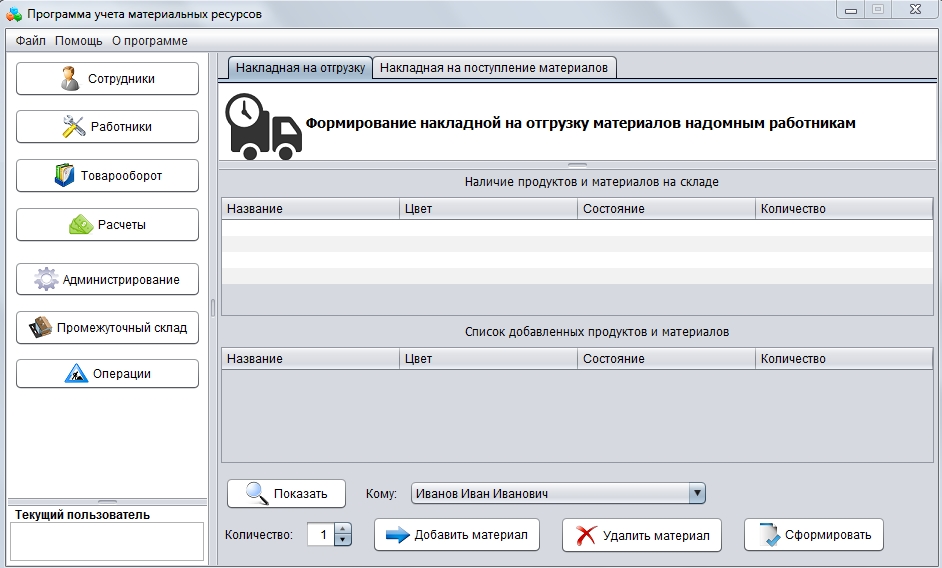


**Рисунок 3.2.2.1 Модуль «Работники»**

### 3.2.3 Модуль «Товарооборот»

Данный модуль позволяет генерировать накладные в формате pdf. Первая вкладка «Накладная на отгрузку» (Рисунок 3.2.3.1 Модуль «Товарооборот»), позволяет на основе имеющихся на складе материалов, осуществлять генерацию накладных на отгрузку товаров со склада.

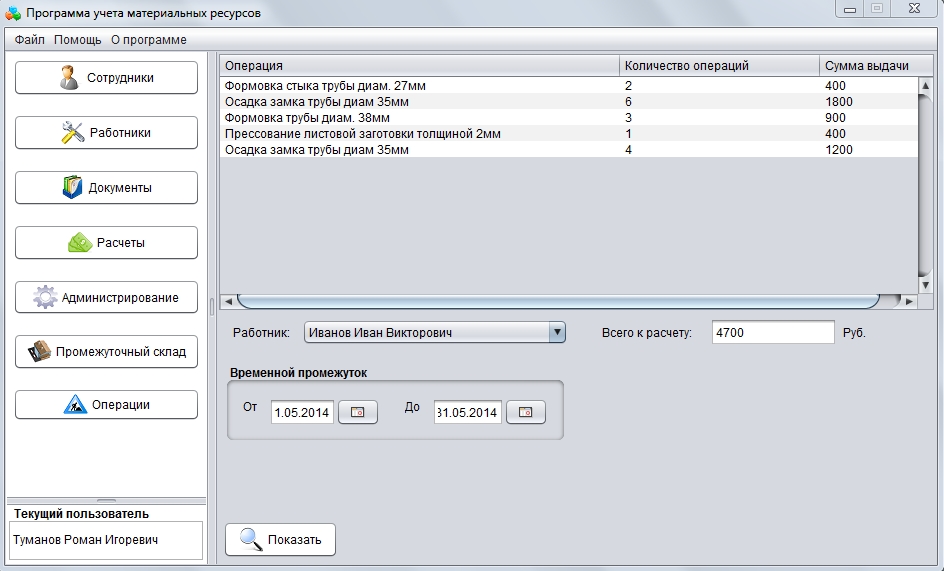
Вторая вкладка «Накладная на поступление материалов», позволяет на основе имеющихся на складе материалов, осуществлять генерацию накладных на поступление материалов на склад.



**Рисунок 3.2.3.1 Модуль «Товарооборот»**

### 3.2.4 Модуль «Расчеты»

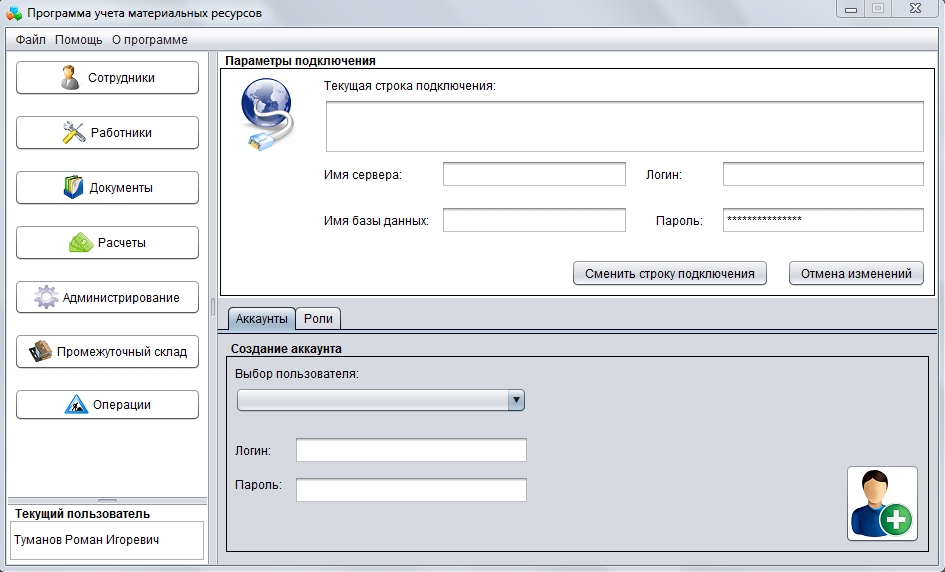
Данный модуль позволяет получать информацию о проделанной работе в указанный промежуток времени, и рассчитывать стоимости проведенных работ (Рисунок 3.2.4.1 Модуль «Расчеты»).



**Рисунок 3.2.4.1 Модуль «Расчеты»**

### 3.2.5 Модуль «Администрирование»

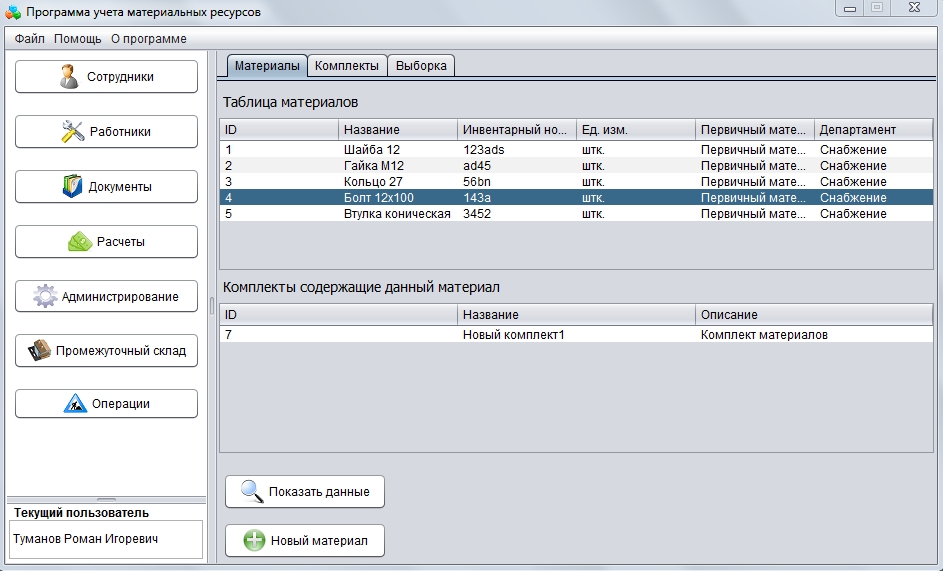
Данный модуль позволяет изменять параметры подключения в БД, задавать пользователям логин и пароль для входа в систему, и назначать пользователям роли (Рисунок 3.2.5.1 Модуль «Администрирование»).



**Рисунок 3.2.5.1 Модуль «Администрирование»**

### 3.2.6 Модуль «Промежуточный склад»

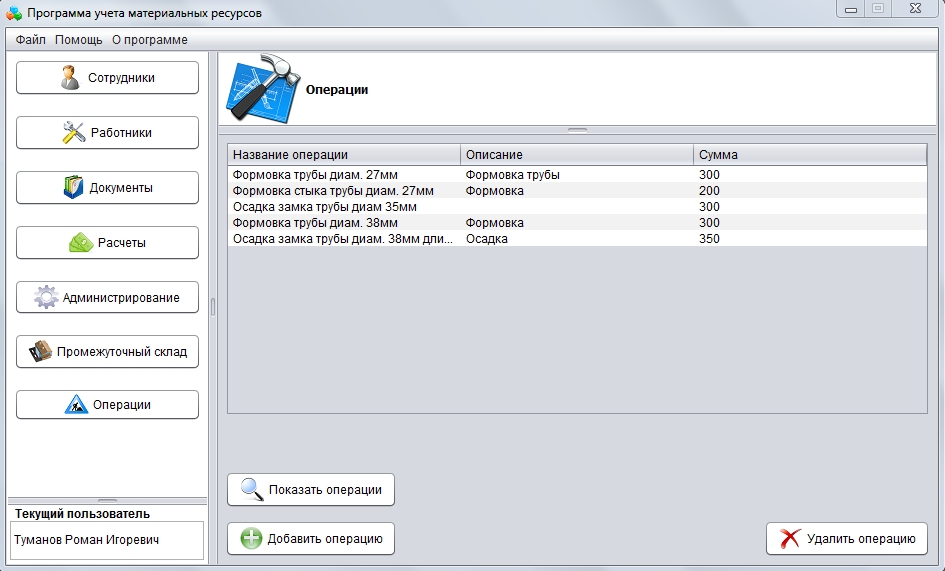
Данный модуль позволяет добавлять материалы в БД, удалять материалы, добавлять и просматривать комплекты материалов, показывать состояние склада (Рисунок 3.2.6.1 Модуль «Промежуточный склад»). Первая вкладка «Материалы» позволяет просматривать все существующие материалы и по каждому материалу выводить список комплектов, которые содержать данный материал. Вторая вкладка «Материалы» позволяет просматривать все существующие комплекты и по каждому комплекту выводить список материалов, которые содержит данный комплект.



**Рисунок 3.2.6.1 Модуль «Промежуточный склад»**

### 3.2.7 Модуль «Операции»

Данный модуль позволяет показывать список доступных для работников операций, добавлять новые, и удалять уже существующие (Рисунок 3.2.7.1 Модуль «Операции»).



**Рисунок 3.2.7.1 Модуль «Операции»**

## 3.3 Описание классов программы

## Классы программы делятся на 3 типа:

## GUI – классы, которые отвечают за интерфейс программы, к ним относятся формы JFrame и JPanel.

## Класс addcomplect.java – класс осуществляющий добавление пользователем комплектов материалов в БД предприятия посредством отдельного окна. Данная операция доступна только администратору системы. Данный класс взаимодействует с классом addmaterial.java, т.к. при добавлении нового комплекта в него добавляется список материалов.

## Класс addmaterial.java – класс осуществляющий добавление материала в БД предприятия посредством отдельного окна. Данная операция доступна только администратору системы. Данный класс так же взаимодействует с классом Complects.java, в объекты которого происходит добавление новых комплектов.

## Класс addoperation.java – класс осуществляющий добавление пользователем операций доступных для работников в БД предприятия посредством отдельного окна. Данный класс взаимодействует так же с классом Products.java, т.к. в объекты этого класса осуществляется запись новых сотрудников. Взаимодействие происходит так же с классом Operations.java.

## Класс addperson.java – класс осуществляющий добавление сотрудников предприятия в БД. Добавление происходит в отдельном окне, в котором все введенные данные проверяются на корректность ввода, в случае ошибок, пользователь будет о них оповещен. Данная операция доступна только администратору системы.

## Класс addwarehouse.java – класс осуществляющий добавление материалов на склад.

## Класс adminpanel.java – класс реализующий модуль программы «Администрирование».

## Класс calculation.java - класс реализующий модуль программы «Расчеты».

## Класс documentpanel.java - класс реализующий модуль программы «Документы».

## Класс login.java - класс реализующий стартовую форму логина и пароля.

## Класс main.java – главный класс приложения, который осуществляет вызовы всех модулей программы, реализует отображение меню программы и весь остальной интерфейс.

## Класс operations.java - класс реализующий модуль программы «Операции».

## Класс peoplepanel.java - класс реализующий модуль программы «Сотрудники».

## Класс warehouse.java - класс реализующий модуль программы «Промежуточный склад».

## Класс workers.java - класс реализующий модуль программы «Работники».

## Классы отвечающие за хранение и обработку таблиц БД, это классы сущности.

## Работа с документами:

## DocumentStates.java – класс хранящий состояния документов в базе данных предприятия.

## InvoiceDetails – класс хранящий подробные данные по каждому документу в БД предприятия.

## InvoiceTypes – класс хранящий типы документов.

## Invoices – класс хранящий все созданные документы.

## Работа с сотрудниками:

## Contacts – класс реализующий соответствующую сущность в БД, отвечает за хранение персональных данных сотрудников предприятия.

## Drivers - класс реализующий соответствующую сущность в БД, отвечает за хранение базы водителей предприятия.

## Users - класс реализующий соответствующую сущность в БД, отвечает за хранение базы пользователей предприятия.

## Workers - класс реализующий соответствующую сущность в БД, отвечает за хранение базы рабочих предприятия.

## Работа с материалами и комплектами:

## Complects - класс реализующий соответствующую сущность в БД, отвечает за хранение базы комплектов предприятия.

## Products - класс реализующий соответствующую сущность в БД, отвечает за хранение базы материалов предприятия.

## ProductsComplects - класс реализующий соответствующую сущность в БД, отвечает за хранение списка материалов каждого созданного комплекта.

## ProductsOperations - класс реализующий соответствующую сущность в БД, отвечает за хранение базы операций над материалами предприятия.

## ProductsStates - класс реализующий соответствующую сущность в БД, отвечает за хранение базы состояний материалов и комплектов предприятия.

## Другое:

## Operations – класс реализующий соответствующую сущность в БД, отвечает за хранение базы операций.

## Warehouse – класс реализующий соответствующую сущность в БД, отвечает за хранение базы склада.

## WorkersOperations – класс реализующий соответствующую сущность в БД, отвечает за хранение базы операций выполненных работниками.

## Файлы осуществляющие взаимодействие классов описанных в предыдущем пункте с базой.

## hibernate.cfg.xml - конфигурационный файл, который осуществляет взаимодействие классов программы с сущностями БД предприятия.

## 3.4 Взаимодействие пользователя с приложением (распределение по ролям)

## Скриншот (21.05.2014 11-27-38).jpgВзаимодействие пользователя с приложением основано на различных правах доступа. Все рассмотренные выше модули были рассмотрены через учетную запись Администратора, которому доступны все модули и все функции программы. При запуске программы пользователь должен ввести свой логии и пароль (Рисунок 3.4. 1 «Вход в программу»).

**Рисунок 3.4.1 Вход в программу**

## Если все было введено, верно, пользователи с разными ролями увидят следующие главные окна программы:

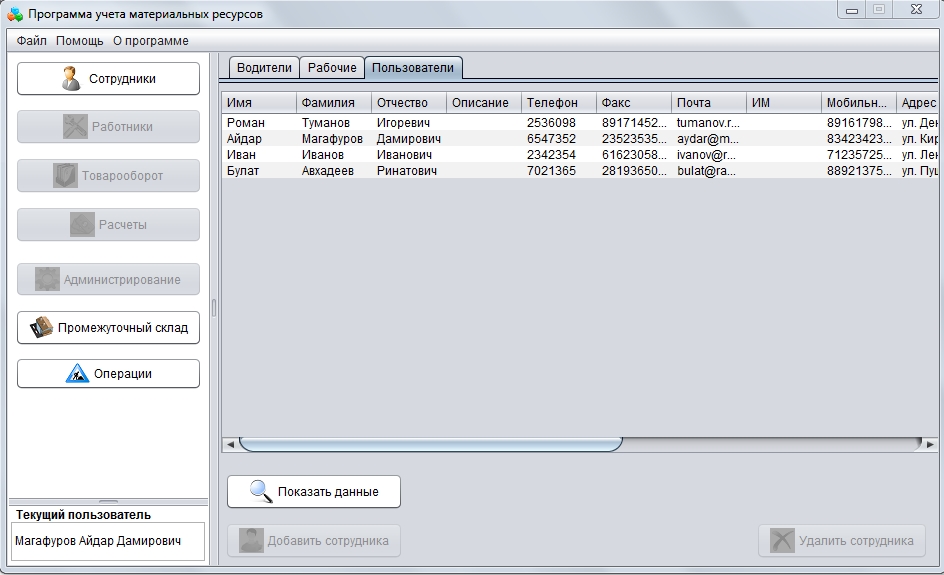
## Администратор: доступны все модули и функции программы, нет никаких ограничений (Рисунок 3.4.2 «Стартовое окно Администратора»).

## Пользователь: самая ограниченная по функционалу роль, доступен только просмотр некоторых данных, без возможности редактирования. Доступны модули: Сотрудники, Промежуточный склад, Операции (Рисунок 3.4.3 «Окно Пользователя»).

## Товаровед: доступно больше модулей и функций чем у Пользователя: Сотрудники, Промежуточный склад, Операции, Товарооборот. Модули Промежуточный склад и Товарооборот предназначены для Товароведа, т.е. у него доступен полный функционал (Рисунок 3.4.4 «Окно Товароведа»). Причем если у пользователя недоступен какой-то элемент, он может увидеть в сплывающей подсказке кому из других пользователей он доступен (Рисунок 3.4.5 «Окно Товароведа»).

## Бухгалтер: роль примерно одно уровня с Товароведом, профильный модуль – Расчеты, который доступен также Администратору (Рисунок 3.4.6 «Окно Бухгалтера»).

## Скриншот (26.05.2014 01-15-00).jpg

**Рисунок 3.4.2 Стартовое окно Администратора**

**Рисунок 3.4.3 Окно Пользователя**

## Скриншот (26.05.2014 01-22-11).jpgРисунок 3.4.4 Окно Товароведа

## Скриншот (26.05.2014 01-22-49).jpg

## Рисунок 3.4.5 Окно Товароведа

## Скриншот (26.05.2014 01-22-11).jpg

## Скриншот (26.05.2014 01-28-03).jpgРисунок 3.4.6 Окно Бухгалтера

# Заключение

## В ходе выполнения данной работы были решены следующие задачи:

1. Проектирование базы данных предприятия;
2. Разработка архитектуры программы;
3. Разработка интерфейса программы;
4. Создание прототипа программы, работа с базой данных;
5. Реализация аутентификации пользователей, создание групп пользователей с разными правами;
6. Разработка модулей программы;
7. Тестирование.

Результатом работы является программа, которая позволяет вести учет ресурсов небольшого территориально-распределенного предприятия с небольшим парком рабочих станций(5-30), используя самостоятельную базу данных. Данная программа учитывает все особенности данного предприятия, имеет дружественный интерфейс, и возможность для быстрой модификации. В программе отсутствует избыточность функционала, которая свойственна решениям, рассмотренным в первой главе.

Данная программа имеет практическую ценность, так как будет использоваться на реальном предприятии. В дальнейшем планируются модификации программы, которые могут потребоваться в ходе работы предприятия. На данный момент программа полностью соответствует требованиям и можно утверждать, что цель данной работы достигнута.

# Список использованных источников

1. Bloch, J. (2008). *Effective Java (2nd Edition)*. Addison-Wesley.
2. Bock, H. (2009). *The Definitive Guide to NetBeans Platform*. Apress.
3. Bauer, Christian, King, Gavin. [*Java Persistence with Hibernate*](http://www.manning.com/bauer2/) / Foreword by Linda DeMichiel. — 1st ed. — Greenwich: [Manning Publications](http://en.wikipedia.org/wiki/Manning_Publications), 2006, November 24. — 904 p. — [ISBN 1-932394-88-5](http://ru.wikipedia.org/wiki/%D0%A1%D0%BB%D1%83%D0%B6%D0%B5%D0%B1%D0%BD%D0%B0%D1%8F:%D0%98%D1%81%D1%82%D0%BE%D1%87%D0%BD%D0%B8%D0%BA%D0%B8_%D0%BA%D0%BD%D0%B8%D0%B3/1932394885), [ISBN 978-1-932394-88-7](http://ru.wikipedia.org/wiki/%D0%A1%D0%BB%D1%83%D0%B6%D0%B5%D0%B1%D0%BD%D0%B0%D1%8F:%D0%98%D1%81%D1%82%D0%BE%D1%87%D0%BD%D0%B8%D0%BA%D0%B8_%D0%BA%D0%BD%D0%B8%D0%B3/9781932394887)
4. Houston, N. (2009). *Microsoft Dynamics for Manufacturing – Understanding the Difference Between GP, NAV, SL and AX*. Retrieved December 5, 2013, http://blog.softwareadvice.com/articles/manufacturing/understanding-the-difference-between-gp-nav-sl-ax-1111709/
5. Morelli, A. (2013). *ERP Fundamentals.* Chmappservices.
6. Нуралиев, С. (2004). *Архитектура "1С: Предприятия" как продукт инженерной мысли*, Просмотрено Ноябрь 13, 2013, официальный сайт 1С:Предприятие: http://v8.1c.ru/metod/architecture/
7. Schwartz, B. & Zaitsev, P. & Tkachenko, V. (2012). *High Performance MySQL, 3rd Edition.* O'Reilly.
8. Varela**,** C**.** A. (2013).*Programming Distributed Computing Systems: A Foundational Approach***.** The MIT Press**.**
9. Weiner, D. (2005). *Dynamics AX: A Guide to Microsoft Axapta*. Apress.