ГОУ ВПО

Уфимский государственный авиационный технический университет

Кафедра вычислительной математики и кибернетики

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | **5** | **6** | **7** | **8** | **9** | **10** | **11** | **12** | **13** | **14** | **15** | **16** |
| **100** |  |  |  |  |  |  |  |  |  |  |  |  |
| **90** |  |  |  |  |  |  |  |  |  |  |  |  |
| **80** |  |  |  |  |  |  |  |  |  |  |  |  |
| **70** |  |  |  |  |  |  |  |  |  |  |  |  |
| **60** |  |  |  |  |  |  |  |  |  |  |  |  |
| **50** |  |  |  |  |  |  |  |  |  |  |  |  |
| **40** |  |  |  |  |  |  |  |  |  |  |  |  |
| **30** |  |  |  |  |  |  |  |  |  |  |  |  |
| **20** |  |  |  |  |  |  |  |  |  |  |  |  |
| **10** |  |  |  |  |  |  |  |  |  |  |  |  |

***Разработка клиент-серверного приложения***

**«Автоматизированная система построения графика работы на площадках добычи нефти»**

**ПОЯСНИТЕЛЬНАЯ ЗАПИСКА**

**к курсовой работе**

**по дисциплине *«Базы данных»***

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Группа | Ф.И.О. | Подпись | Дата | Оценка |
| Студент | Доронин С.Г. |  |  |  |
| Консультант |  |  |  |  |
| Принял |  |  |  |  |

Уфа - 2012 г.

Федеральное агентство по образованию

Государственное образовательное учреждение высшего профессионального образования

«УФИМСКИЙ ГОСУДАРСТВЕННЫЙ АВИАЦИОННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»

Кафедра вычислительной математики и кибернетики

ЗАДАНИЕ

На курсовую работу по дисциплине «Базы данных»

Студент \_\_Доронин С.Г. Группа \_ПРО-201в Консультант\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Ф.И.О номер акад. гр. Ф.И.О. .

1. Тема курсовой работы

Автоматизированная система построения графика работы на площадках добычи нефти

|  |
| --- |
| наименование темы |
|  |

|  |  |
| --- | --- |
| 1. Основное содержание: | Разработка информационной модели предметной области, |
| разработка программного продукта для работы с клиент-серверной СУБД, создание | |
| программной документации | |

1. Требования к оформлению
   1. Техническое задание для программного продукта должно быть оформлено согласно ГОСТу 19.201-78 «Техническое задание. Требования к содержанию и оформлению».
   2. Руководство пользователя (оператора) разрабатывается и оформляется согласно ГОСТу19.505-79 «Руководство оператора. Требования к содержанию и оформлению».
   3. В пояснительной записке должны содержаться следующие разделы:

* Введение – в этой части описывается цель курсовой работы, требования к разрабатываемой базе данных и приложению.
* Техническое задание, оформленное согласно ГОСТу.
* ER-диаграмма (логический и физический уровень).
* Программная реализация ER-диаграммы (с комментариями SQL-кода).
* Разработка приложения – в этой части должна быть представлена структура модулей программы и их взаимосвязь, описана схема работы с базой данных на уровне компонентов, описаны особенности реализации приложения.
* Руководство пользователя, оформленное согласно ГОСТу.
* Заключение – эта часть содержит описание основных результатов работы, сферу его применения.

Дата выдачи \_\_\_\_\_\_\_\_\_\_\_\_ Дата окончания \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Руководитель \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Зотова О.Ф.

Подпись

# Введение

В ЭГТИ используется большое количество площадок и рабочих для добычи нефти, в связи с этим для эффективной деятельности требуется построение графика работы сотрудников на площадках.

В данный момент график строится одним человеком вручную в Excel. Это рутинная работа, которая подвержена ошибкам, так как приходится учитывать много условий. Например, находится ли сотрудник в отпуске или он уволен, или не прошло определенное количество дней с его последнего приезда и т.д.

В среднем на это уходит 2-3 дня, а это солидное время, которое может быть потрачено на более важные дела.

Кроме того, в конце месяца нужно составить отчеты о работе сотрудников. Это тоже делается вручную в Excel. Это еще более рутинная работа, которой занимается несколько человек, потому что в организации большой штат сотрудников.

На составление отчетов в среднем уходит 7-8 дней, к тому же приходится отвлекать бухгалтерию от их основной работы.

Сейчас все это можно автоматизировать. Например, уже решены некоторые подобные задачи контроля расходов и управления услугами через Интернет (все сотовые операторы используют веб-приложения для того этого).

Новизна данного программного решения состоит в том, что в нём реализована удобная система построения графика работы сотрудников и автоматизированное составление отчетов на основе графика.

Данное программное решение предназначено для автоматизации построения графика работы сотрудников на площадках добычи нефти. Предлагаемый проект позволит пользователю намного облегчить его работу, более быстро осуществлять ввод, поиск и корректировку данных. А так же автоматизировано составлять отчеты о работе сотрудников.

Под автоматизированной системой построения графика работы на площадках добычи нефти будем понимать программный комплекс, решающий эту задачу.

ГОУ ВПО

Уфимский государственный авиационный технический университет

Кафедра вычислительной математики и кибернетики

**Программный продукт**

**«Автоматизированная система построения графика работы на площадках добычи нефти»**

# Техническое задание

**Листов 6**

2012

### Аннотация

Программный продукт (ПП) «Автоматизированная система построения графика работы на площадках добычи нефти» предназначен для построения графика работы сотрудников на площадках добычи нефти.

Кроме того, программа может применяться для построения табелей работы сотрудников по областям.

Данный документ описывает:

* наименование, краткую характеристику области применения программного продукта;
* основание для разработки программного продукта;
* назначение разработки, в котором указано функциональное назначение программного продукта;
* требования к программному продукту;
* состав программной документации.

### Введение

«Автоматизированная система построения графика работы на площадках добычи нефти» предназначена для построения графика работы сотрудников на площадках добычи нефти, а также автоматизированного построения отчетов о работе сотрудников.

### Основание для разработки

Вышеназванный программный продукт разрабатывается на основании потребностей ЭГТИ.

### Назначение разработки

ПП должен состоять из следующих функциональных модулей:

1. главный модуль, обеспечивающий добавление, удаление и редактирование работы сотрудников на графике, а также совместную работу всех модулей;
2. модуль ввода различных данных для создания графика и отчетности;
3. модуль импорта сотрудников из XLS-файла, обеспечивает автоматизированное управление сотрудниками;
4. модуль построения табелей по областям, где работали сотрудники;
5. модуль хранения настроек приложения;
6. модуль хранения настроек для построения отчетности;
7. модуль авторизации, обеспечивает защиту данных от неавторизованных пользователей.

### Требования к программе

#### Требования к функциональным характеристикам

1. Модуль ввода различных данных для создания графика и отчетности должен обеспечить добавление, удаление, редактирование, сохранение в БД и проверку корректности введенных данных для:
   * + типа дней (название, цвет на графике, описание);
     + областей, на которых находятся площадки (название, описание, название листа в отчете, заголовок в отчете);
     + название, описание
     + площадок, на которых производятся работы (название, описание, область, тип площадки, номер площадки, № телефона, заказчик, тип вагона, тип оборудования);
     + (название, описание)
     + (название, описание)
     + сотрудников (табельный №, подразделение, должность, фамилия, имя, отчество, дата рождения, дата приема на работу, образование, ИНН, № паспорта, пенсионный №, адрес по прописке, заработная плата);
     + рабочих часов (количество часов, краткое описание, описание);
     + нормы выработки за месяц для сотрудников (год, январь, февраль, март и т.д.);
     + праздничных дней (дата).
2. г (тип дня, сотрудник, площадка, количество часов, дата начала, дата конца, № сотрудника)
3. модуль импорта сотрудников из XLS-файла должен обеспечить выполнение следующих функций:
   * + автоматизированное добавление новых сотрудников в БД;
     + вывод в виде списка уже существующих сотрудников;
     + вывод в виде списка сотрудников с частью различных данных (для ручной правки).
4. модуль построения табелей по областям, где работали сотрудники, должен обеспечить:
   * + автоматизированное создание табелей по всем областям, где работали сотрудники в XLS-файл на нескольких листах (каждая область на новом листе).
5. модуль хранения настроек приложения должен обеспечить:
   * + изменение настроек приложения без его перезапуска;
     + хранение настроек приложения в БД (дата последнего запуска приложения).
6. модуль хранения настроек для построения отчетности должен обеспечить:
   * + изменение настроек построения отчетности без перезапуска приложения;
     + хранения настроек для отчетности в БД (стоимость отпускного дня, размер премии в %).
7. модуль авторизации должен обеспечить:
   * + (название роли, описание)
     + возможность запуска приложения только зарегистрированным пользователям, имеющим логин и пароль (имя, логин, пароль, роль);
     + возможность изменения, удаления и добавления данных только пользователям с ролью «Администратор».

#### Требования к надежности

«Автоматизированная система построения графика работы на площадках добычи нефти» должна обеспечивать следующие требования к надежности:

* устойчивая работа системы в целом;
* контроль на ошибочность вводимых данных;
* проверка на допустимость числовых данных.

#### Требования к составу и параметрам технических средств

К разрабатываемому программному продукту предъявляются следующие минимальные аппаратные требования:

Сервер:

1. IBM PC-совместимый компьютер с процессором – Intel® Pentium® IV или выше;
2. оперативная память – не менее 2048Mb;
3. операционная система Microsoft® Windows® XP (Vista, 7, 8), Linux, MacOS;
4. веб-сервер Apache Tomcat 7;
5. сервер баз данных MySQL (MS SQL, DB2, PostgreSQL).

Клиент:

1. IBM PC-совместимый компьютер с процессором – Intel® Pentium® III или выше;
2. оперативная память – не менее 512Mb;
3. SVGA-монитор с разрешением не ниже 1280x1024;
4. клавиатура;
5. манипулятор типа «мышь»;
6. веб-браузер Google Chrome, Opera, Firefox, IE (не ниже 10 версии);
7. операционная система Microsoft® Windows® XP (Vista, 7, 8), Linux, MacOS.

#### Требования к информационной и программной совместимости

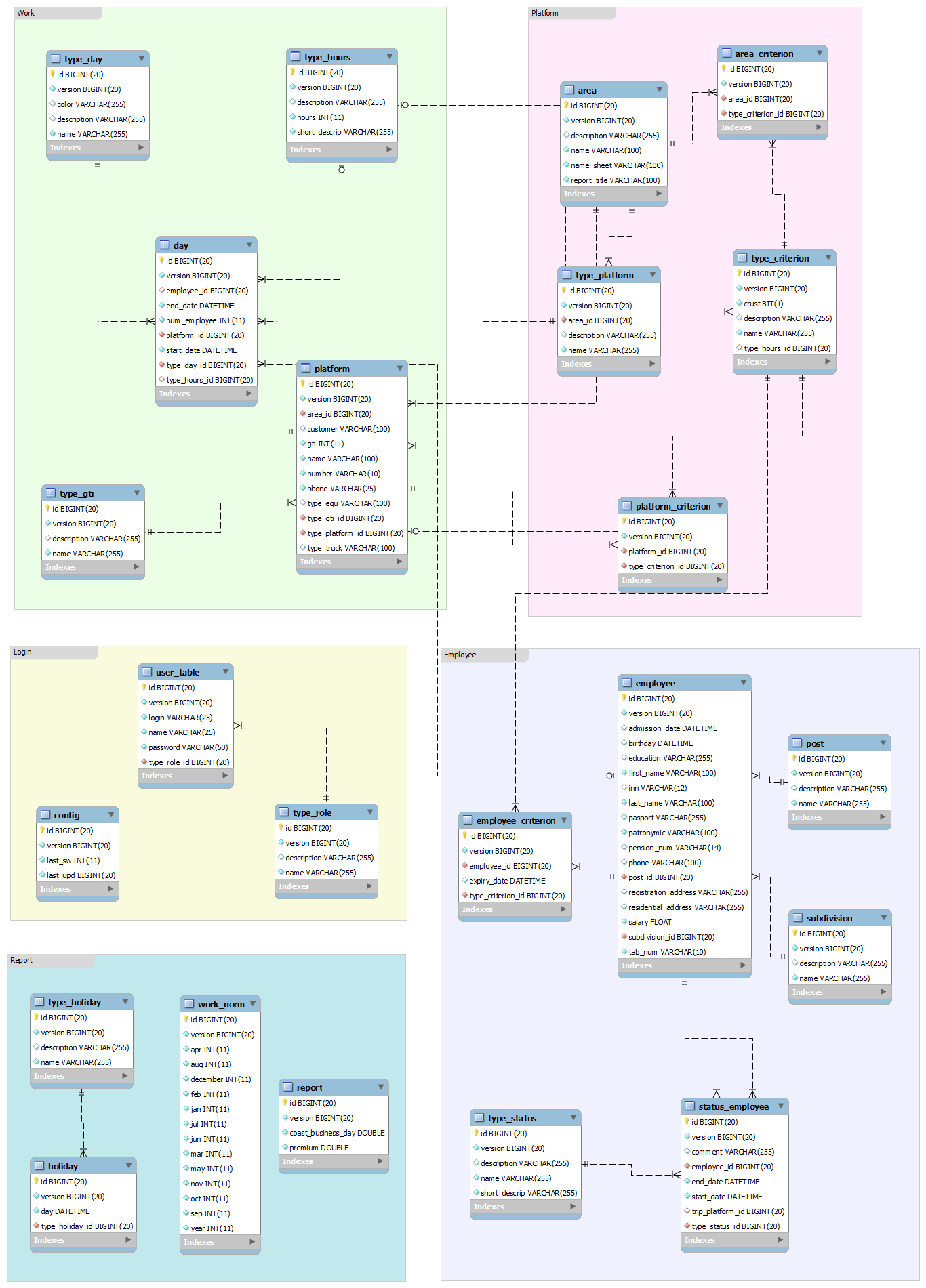
Часть входных данных вводится вручную, данные берутся из документов. Часть данных импортируется из XLS-файла, который берется из «1C-Бухгалтерия». Выходные данные сохраняются в виде XLS-файла.

Программа должна быть реализована на фреймворке Grails 2.0.4 в интегрированной среде разработки IntelliJ IDEA 11 с возможностью дальнейшего расширения и переработки с использованием этой среды или ее более поздних версий.

### Требования к программной документации

Документация на программный продукт должна включать в себя следующие документы: техническое задание, руководство пользователя.

# ER-диаграмма (логический и физический уровень)



# Программная реализация ER-диаграммы (с комментариями SQL-кода)

--

-- Создание БД "tabel" если ее не существует

--

CREATE DATABASE IF NOT EXISTS `tabel` /\*!40100 DEFAULT CHARACTER SET utf8 \*/;

USE `tabel`;

-- MySQL dump 10.13 Distrib 5.5.16, for Win32 (x86)

--

-- Host: localhost Database: tabel

-- ------------------------------------------------------

-- Server version 5.5.28-log

/\*!40101 SET @OLD\_CHARACTER\_SET\_CLIENT=@@CHARACTER\_SET\_CLIENT \*/;

/\*!40101 SET @OLD\_CHARACTER\_SET\_RESULTS=@@CHARACTER\_SET\_RESULTS \*/;

/\*!40101 SET @OLD\_COLLATION\_CONNECTION=@@COLLATION\_CONNECTION \*/;

/\*!40101 SET NAMES utf8 \*/;

/\*!40103 SET @OLD\_TIME\_ZONE=@@TIME\_ZONE \*/;

/\*!40103 SET TIME\_ZONE='+00:00' \*/;

/\*!40014 SET @OLD\_UNIQUE\_CHECKS=@@UNIQUE\_CHECKS, UNIQUE\_CHECKS=0 \*/;

/\*!40014 SET @OLD\_FOREIGN\_KEY\_CHECKS=@@FOREIGN\_KEY\_CHECKS, FOREIGN\_KEY\_CHECKS=0 \*/;

/\*!40101 SET @OLD\_SQL\_MODE=@@SQL\_MODE, SQL\_MODE='NO\_AUTO\_VALUE\_ON\_ZERO' \*/;

/\*!40111 SET @OLD\_SQL\_NOTES=@@SQL\_NOTES, SQL\_NOTES=0 \*/;

--

-- Создание таблицы "Нормы выработки"

--

DROP TABLE IF EXISTS `work\_norm`;

/\*!40101 SET @saved\_cs\_client = @@character\_set\_client \*/;

/\*!40101 SET character\_set\_client = utf8 \*/;

CREATE TABLE `work\_norm` (

`id` bigint(20) NOT NULL AUTO\_INCREMENT,

`version` bigint(20) NOT NULL,

`apr` int(11) NOT NULL,

`aug` int(11) NOT NULL,

`december` int(11) NOT NULL,

`feb` int(11) NOT NULL,

`jan` int(11) NOT NULL,

`jul` int(11) NOT NULL,

`jun` int(11) NOT NULL,

`mar` int(11) NOT NULL,

`may` int(11) NOT NULL,

`nov` int(11) NOT NULL,

`oct` int(11) NOT NULL,

`sep` int(11) NOT NULL,

`year` int(11) NOT NULL,

PRIMARY KEY (`id`)

) ENGINE=InnoDB AUTO\_INCREMENT=2 DEFAULT CHARSET=utf8;

/\*!40101 SET character\_set\_client = @saved\_cs\_client \*/;

--

-- Создание таблицы "Типы праздничных дней"

--

DROP TABLE IF EXISTS `type\_holiday`;

/\*!40101 SET @saved\_cs\_client = @@character\_set\_client \*/;

/\*!40101 SET character\_set\_client = utf8 \*/;

CREATE TABLE `type\_holiday` (

`id` bigint(20) NOT NULL,

`version` bigint(20) NOT NULL,

`description` varchar(255) DEFAULT NULL,

`name` varchar(255) NOT NULL,

PRIMARY KEY (`id`)

) ENGINE=InnoDB DEFAULT CHARSET=utf8;

/\*!40101 SET character\_set\_client = @saved\_cs\_client \*/;

--

-- Создание таблицы "Типы платформ"

--

DROP TABLE IF EXISTS `type\_platform`;

/\*!40101 SET @saved\_cs\_client = @@character\_set\_client \*/;

/\*!40101 SET character\_set\_client = utf8 \*/;

CREATE TABLE `type\_platform` (

`id` bigint(20) NOT NULL AUTO\_INCREMENT,

`version` bigint(20) NOT NULL,

`area\_id` bigint(20) NOT NULL,

`description` varchar(255) DEFAULT NULL,

`name` varchar(255) NOT NULL,

PRIMARY KEY (`id`),

KEY `FKB0808BD86D389962` (`area\_id`),

CONSTRAINT `FKB0808BD86D389962` FOREIGN KEY (`area\_id`) REFERENCES `area` (`id`)

) ENGINE=InnoDB AUTO\_INCREMENT=6 DEFAULT CHARSET=utf8;

/\*!40101 SET character\_set\_client = @saved\_cs\_client \*/;

--

-- Создание таблицы "Пользователи"

--

DROP TABLE IF EXISTS `user\_table`;

/\*!40101 SET @saved\_cs\_client = @@character\_set\_client \*/;

/\*!40101 SET character\_set\_client = utf8 \*/;

CREATE TABLE `user\_table` (

`id` bigint(20) NOT NULL AUTO\_INCREMENT,

`version` bigint(20) NOT NULL,

`login` varchar(25) NOT NULL,

`name` varchar(25) NOT NULL,

`password` varchar(50) NOT NULL,

`type\_role\_id` bigint(20) NOT NULL,

PRIMARY KEY (`id`),

UNIQUE KEY `login` (`login`),

UNIQUE KEY `name` (`name`),

KEY `FKC6DD463AF58AAAF5` (`type\_role\_id`),

CONSTRAINT `FKC6DD463AF58AAAF5` FOREIGN KEY (`type\_role\_id`) REFERENCES `type\_role` (`id`)

) ENGINE=InnoDB AUTO\_INCREMENT=3 DEFAULT CHARSET=utf8;

/\*!40101 SET character\_set\_client = @saved\_cs\_client \*/;

--

-- Создание таблицы "Типы ролей"

--

DROP TABLE IF EXISTS `type\_role`;

/\*!40101 SET @saved\_cs\_client = @@character\_set\_client \*/;

/\*!40101 SET character\_set\_client = utf8 \*/;

CREATE TABLE `type\_role` (

`id` bigint(20) NOT NULL,

`version` bigint(20) NOT NULL,

`description` varchar(255) DEFAULT NULL,

`name` varchar(255) NOT NULL,

PRIMARY KEY (`id`),

UNIQUE KEY `name` (`name`)

) ENGINE=InnoDB DEFAULT CHARSET=utf8;

/\*!40101 SET character\_set\_client = @saved\_cs\_client \*/;

--

-- Создание таблицы "Типы дней"

--

DROP TABLE IF EXISTS `type\_day`;

/\*!40101 SET @saved\_cs\_client = @@character\_set\_client \*/;

/\*!40101 SET character\_set\_client = utf8 \*/;

CREATE TABLE `type\_day` (

`id` bigint(20) NOT NULL,

`version` bigint(20) NOT NULL,

`color` varchar(255) DEFAULT NULL,

`description` varchar(255) DEFAULT NULL,

`name` varchar(255) NOT NULL,

PRIMARY KEY (`id`)

) ENGINE=InnoDB DEFAULT CHARSET=utf8;

/\*!40101 SET character\_set\_client = @saved\_cs\_client \*/;

--

-- Создание таблицы "Настройки для отчетов"

--

DROP TABLE IF EXISTS `report`;

/\*!40101 SET @saved\_cs\_client = @@character\_set\_client \*/;

/\*!40101 SET character\_set\_client = utf8 \*/;

CREATE TABLE `report` (

`id` bigint(20) NOT NULL,

`version` bigint(20) NOT NULL,

`coast\_business\_day` double NOT NULL,

`premium` double NOT NULL,

PRIMARY KEY (`id`)

) ENGINE=InnoDB DEFAULT CHARSET=utf8;

/\*!40101 SET character\_set\_client = @saved\_cs\_client \*/;

--

-- Создание таблицы "Должности"

--

DROP TABLE IF EXISTS `post`;

/\*!40101 SET @saved\_cs\_client = @@character\_set\_client \*/;

/\*!40101 SET character\_set\_client = utf8 \*/;

CREATE TABLE `post` (

`id` bigint(20) NOT NULL AUTO\_INCREMENT,

`version` bigint(20) NOT NULL,

`description` varchar(255) DEFAULT NULL,

`name` varchar(255) NOT NULL,

PRIMARY KEY (`id`)

) ENGINE=InnoDB AUTO\_INCREMENT=3 DEFAULT CHARSET=utf8;

/\*!40101 SET character\_set\_client = @saved\_cs\_client \*/;

--

-- Создание таблицы "Критерии для области"

--

DROP TABLE IF EXISTS `area\_criterion`;

/\*!40101 SET @saved\_cs\_client = @@character\_set\_client \*/;

/\*!40101 SET character\_set\_client = utf8 \*/;

CREATE TABLE `area\_criterion` (

`id` bigint(20) NOT NULL AUTO\_INCREMENT,

`version` bigint(20) NOT NULL,

`area\_id` bigint(20) NOT NULL,

`type\_criterion\_id` bigint(20) NOT NULL,

PRIMARY KEY (`id`),

KEY `FKFD434D6F6D389962` (`area\_id`),

KEY `FKFD434D6FE70B7B7` (`type\_criterion\_id`),

CONSTRAINT `FKFD434D6FE70B7B7` FOREIGN KEY (`type\_criterion\_id`) REFERENCES `type\_criterion` (`id`),

CONSTRAINT `FKFD434D6F6D389962` FOREIGN KEY (`area\_id`) REFERENCES `area` (`id`)

) ENGINE=InnoDB AUTO\_INCREMENT=4 DEFAULT CHARSET=utf8;

/\*!40101 SET character\_set\_client = @saved\_cs\_client \*/;

--

-- Создание таблицы "Критерии для площадок"

--

DROP TABLE IF EXISTS `platform\_criterion`;

/\*!40101 SET @saved\_cs\_client = @@character\_set\_client \*/;

/\*!40101 SET character\_set\_client = utf8 \*/;

CREATE TABLE `platform\_criterion` (

`id` bigint(20) NOT NULL AUTO\_INCREMENT,

`version` bigint(20) NOT NULL,

`platform\_id` bigint(20) NOT NULL,

`type\_criterion\_id` bigint(20) NOT NULL,

PRIMARY KEY (`id`),

KEY `FK30FABDD5E70B7B7` (`type\_criterion\_id`),

KEY `FK30FABDD537B123A2` (`platform\_id`),

CONSTRAINT `FK30FABDD537B123A2` FOREIGN KEY (`platform\_id`) REFERENCES `platform` (`id`),

CONSTRAINT `FK30FABDD5E70B7B7` FOREIGN KEY (`type\_criterion\_id`) REFERENCES `type\_criterion` (`id`)

) ENGINE=InnoDB AUTO\_INCREMENT=4 DEFAULT CHARSET=utf8;

/\*!40101 SET character\_set\_client = @saved\_cs\_client \*/;

--

-- Создание таблицы "Подразделения"

--

DROP TABLE IF EXISTS `subdivision`;

/\*!40101 SET @saved\_cs\_client = @@character\_set\_client \*/;

/\*!40101 SET character\_set\_client = utf8 \*/;

CREATE TABLE `subdivision` (

`id` bigint(20) NOT NULL AUTO\_INCREMENT,

`version` bigint(20) NOT NULL,

`description` varchar(255) DEFAULT NULL,

`name` varchar(255) NOT NULL,

PRIMARY KEY (`id`)

) ENGINE=InnoDB AUTO\_INCREMENT=4 DEFAULT CHARSET=utf8;

/\*!40101 SET character\_set\_client = @saved\_cs\_client \*/;

--

-- Создание таблицы "Типы часов"

--

DROP TABLE IF EXISTS `type\_hours`;

/\*!40101 SET @saved\_cs\_client = @@character\_set\_client \*/;

/\*!40101 SET character\_set\_client = utf8 \*/;

CREATE TABLE `type\_hours` (

`id` bigint(20) NOT NULL,

`version` bigint(20) NOT NULL,

`description` varchar(255) DEFAULT NULL,

`hours` int(11) NOT NULL,

`short\_descrip` varchar(255) NOT NULL,

PRIMARY KEY (`id`)

) ENGINE=InnoDB DEFAULT CHARSET=utf8;

/\*!40101 SET character\_set\_client = @saved\_cs\_client \*/;

--

-- Создание таблицы "Области"

--

DROP TABLE IF EXISTS `area`;

/\*!40101 SET @saved\_cs\_client = @@character\_set\_client \*/;

/\*!40101 SET character\_set\_client = utf8 \*/;

CREATE TABLE `area` (

`id` bigint(20) NOT NULL,

`version` bigint(20) NOT NULL,

`description` varchar(255) DEFAULT NULL,

`name` varchar(100) NOT NULL,

`name\_sheet` varchar(100) NOT NULL,

`report\_title` varchar(100) NOT NULL,

PRIMARY KEY (`id`)

) ENGINE=InnoDB DEFAULT CHARSET=utf8;

/\*!40101 SET character\_set\_client = @saved\_cs\_client \*/;

--

-- Создание таблицы "Критерии для сотрудников"

--

DROP TABLE IF EXISTS `employee\_criterion`;

/\*!40101 SET @saved\_cs\_client = @@character\_set\_client \*/;

/\*!40101 SET character\_set\_client = utf8 \*/;

CREATE TABLE `employee\_criterion` (

`id` bigint(20) NOT NULL AUTO\_INCREMENT,

`version` bigint(20) NOT NULL,

`employee\_id` bigint(20) NOT NULL,

`expiry\_date` datetime DEFAULT NULL,

`type\_criterion\_id` bigint(20) NOT NULL,

PRIMARY KEY (`id`),

KEY `FKAECFEE50E70B7B7` (`type\_criterion\_id`),

KEY `FKAECFEE50D55C3919` (`employee\_id`),

CONSTRAINT `FKAECFEE50D55C3919` FOREIGN KEY (`employee\_id`) REFERENCES `employee` (`id`),

CONSTRAINT `FKAECFEE50E70B7B7` FOREIGN KEY (`type\_criterion\_id`) REFERENCES `type\_criterion` (`id`)

) ENGINE=InnoDB AUTO\_INCREMENT=8 DEFAULT CHARSET=utf8;

/\*!40101 SET character\_set\_client = @saved\_cs\_client \*/;

--

-- Создание таблицы "Рабочие дни"

--

DROP TABLE IF EXISTS `day`;

/\*!40101 SET @saved\_cs\_client = @@character\_set\_client \*/;

/\*!40101 SET character\_set\_client = utf8 \*/;

CREATE TABLE `day` (

`id` bigint(20) NOT NULL AUTO\_INCREMENT,

`version` bigint(20) NOT NULL,

`employee\_id` bigint(20) DEFAULT NULL,

`end\_date` datetime NOT NULL,

`num\_employee` int(11) NOT NULL,

`platform\_id` bigint(20) NOT NULL,

`start\_date` datetime NOT NULL,

`type\_day\_id` bigint(20) NOT NULL,

`type\_hours\_id` bigint(20) DEFAULT NULL,

PRIMARY KEY (`id`),

KEY `FK1839C4F77B02F` (`type\_day\_id`),

KEY `FK1839CCEE3EE0F` (`type\_hours\_id`),

KEY `FK1839C37B123A2` (`platform\_id`),

KEY `FK1839CD55C3919` (`employee\_id`),

CONSTRAINT `FK1839CD55C3919` FOREIGN KEY (`employee\_id`) REFERENCES `employee` (`id`),

CONSTRAINT `FK1839C37B123A2` FOREIGN KEY (`platform\_id`) REFERENCES `platform` (`id`),

CONSTRAINT `FK1839C4F77B02F` FOREIGN KEY (`type\_day\_id`) REFERENCES `type\_day` (`id`),

CONSTRAINT `FK1839CCEE3EE0F` FOREIGN KEY (`type\_hours\_id`) REFERENCES `type\_hours` (`id`)

) ENGINE=InnoDB AUTO\_INCREMENT=6 DEFAULT CHARSET=utf8;

/\*!40101 SET character\_set\_client = @saved\_cs\_client \*/;

--

-- Создание таблицы "Площадки"

--

DROP TABLE IF EXISTS `platform`;

/\*!40101 SET @saved\_cs\_client = @@character\_set\_client \*/;

/\*!40101 SET character\_set\_client = utf8 \*/;

CREATE TABLE `platform` (

`id` bigint(20) NOT NULL AUTO\_INCREMENT,

`version` bigint(20) NOT NULL,

`area\_id` bigint(20) NOT NULL,

`customer` varchar(100) DEFAULT NULL,

`gti` int(11) NOT NULL,

`name` varchar(100) NOT NULL,

`number` varchar(10) NOT NULL,

`phone` varchar(25) NOT NULL,

`type\_equ` varchar(100) DEFAULT NULL,

`type\_gti\_id` bigint(20) NOT NULL,

`type\_platform\_id` bigint(20) NOT NULL,

`type\_truck` varchar(100) DEFAULT NULL,

PRIMARY KEY (`id`),

KEY `FK6FBD68736D389962` (`area\_id`),

KEY `FK6FBD68732757F77` (`type\_platform\_id`),

KEY `FK6FBD687319B577D` (`type\_gti\_id`),

CONSTRAINT `FK6FBD687319B577D` FOREIGN KEY (`type\_gti\_id`) REFERENCES `type\_gti` (`id`),

CONSTRAINT `FK6FBD68732757F77` FOREIGN KEY (`type\_platform\_id`) REFERENCES `type\_platform` (`id`),

CONSTRAINT `FK6FBD68736D389962` FOREIGN KEY (`area\_id`) REFERENCES `area` (`id`)

) ENGINE=InnoDB AUTO\_INCREMENT=5 DEFAULT CHARSET=utf8;

/\*!40101 SET character\_set\_client = @saved\_cs\_client \*/;

--

-- Создание таблицы "Занятость сотрудников"

--

DROP TABLE IF EXISTS `status\_employee`;

/\*!40101 SET @saved\_cs\_client = @@character\_set\_client \*/;

/\*!40101 SET character\_set\_client = utf8 \*/;

CREATE TABLE `status\_employee` (

`id` bigint(20) NOT NULL AUTO\_INCREMENT,

`version` bigint(20) NOT NULL,

`comment` varchar(255) DEFAULT NULL,

`employee\_id` bigint(20) NOT NULL,

`end\_date` datetime NOT NULL,

`start\_date` datetime NOT NULL,

`trip\_platform\_id` bigint(20) DEFAULT NULL,

`type\_status\_id` bigint(20) NOT NULL,

PRIMARY KEY (`id`),

KEY `FKA840185B658267EE` (`type\_status\_id`),

KEY `FKA840185B49B5E688` (`trip\_platform\_id`),

KEY `FKA840185BD55C3919` (`employee\_id`),

CONSTRAINT `FKA840185BD55C3919` FOREIGN KEY (`employee\_id`) REFERENCES `employee` (`id`),

CONSTRAINT `FKA840185B49B5E688` FOREIGN KEY (`trip\_platform\_id`) REFERENCES `platform` (`id`),

CONSTRAINT `FKA840185B658267EE` FOREIGN KEY (`type\_status\_id`) REFERENCES `type\_status` (`id`)

) ENGINE=InnoDB AUTO\_INCREMENT=7 DEFAULT CHARSET=utf8;

/\*!40101 SET character\_set\_client = @saved\_cs\_client \*/;

--

-- Создание таблицы "Типы занятости"

--

DROP TABLE IF EXISTS `type\_status`;

/\*!40101 SET @saved\_cs\_client = @@character\_set\_client \*/;

/\*!40101 SET character\_set\_client = utf8 \*/;

CREATE TABLE `type\_status` (

`id` bigint(20) NOT NULL,

`version` bigint(20) NOT NULL,

`description` varchar(255) DEFAULT NULL,

`name` varchar(255) NOT NULL,

`short\_descrip` varchar(255) NOT NULL,

PRIMARY KEY (`id`)

) ENGINE=InnoDB DEFAULT CHARSET=utf8;

/\*!40101 SET character\_set\_client = @saved\_cs\_client \*/;

--

-- Создание таблицы "Типы ГТИ"

--

DROP TABLE IF EXISTS `type\_gti`;

/\*!40101 SET @saved\_cs\_client = @@character\_set\_client \*/;

/\*!40101 SET character\_set\_client = utf8 \*/;

CREATE TABLE `type\_gti` (

`id` bigint(20) NOT NULL AUTO\_INCREMENT,

`version` bigint(20) NOT NULL,

`description` varchar(255) DEFAULT NULL,

`name` varchar(255) NOT NULL,

PRIMARY KEY (`id`)

) ENGINE=InnoDB AUTO\_INCREMENT=3 DEFAULT CHARSET=utf8;

/\*!40101 SET character\_set\_client = @saved\_cs\_client \*/;

--

-- Создание таблицы "Сотрудники"

--

DROP TABLE IF EXISTS `employee`;

/\*!40101 SET @saved\_cs\_client = @@character\_set\_client \*/;

/\*!40101 SET character\_set\_client = utf8 \*/;

CREATE TABLE `employee` (

`id` bigint(20) NOT NULL AUTO\_INCREMENT,

`version` bigint(20) NOT NULL,

`admission\_date` datetime DEFAULT NULL,

`birthday` datetime DEFAULT NULL,

`education` varchar(255) DEFAULT NULL,

`first\_name` varchar(100) NOT NULL,

`inn` varchar(12) DEFAULT NULL,

`last\_name` varchar(100) NOT NULL,

`pasport` varchar(255) DEFAULT NULL,

`patronymic` varchar(100) NOT NULL,

`pension\_num` varchar(14) DEFAULT NULL,

`phone` varchar(100) DEFAULT NULL,

`post\_id` bigint(20) NOT NULL,

`registration\_address` varchar(255) DEFAULT NULL,

`residential\_address` varchar(255) DEFAULT NULL,

`salary` float NOT NULL,

`subdivision\_id` bigint(20) NOT NULL,

`tab\_num` varchar(10) NOT NULL,

PRIMARY KEY (`id`),

UNIQUE KEY `tab\_num` (`tab\_num`),

KEY `FK4722E6AE58946E84` (`subdivision\_id`),

KEY `FK4722E6AE79928359` (`post\_id`),

CONSTRAINT `FK4722E6AE79928359` FOREIGN KEY (`post\_id`) REFERENCES `post` (`id`),

CONSTRAINT `FK4722E6AE58946E84` FOREIGN KEY (`subdivision\_id`) REFERENCES `subdivision` (`id`)

) ENGINE=InnoDB AUTO\_INCREMENT=5 DEFAULT CHARSET=utf8;

/\*!40101 SET character\_set\_client = @saved\_cs\_client \*/;

--

-- Создание таблицы "Праздничные дни"

--

DROP TABLE IF EXISTS `holiday`;

/\*!40101 SET @saved\_cs\_client = @@character\_set\_client \*/;

/\*!40101 SET character\_set\_client = utf8 \*/;

CREATE TABLE `holiday` (

`id` bigint(20) NOT NULL AUTO\_INCREMENT,

`version` bigint(20) NOT NULL,

`day` datetime NOT NULL,

`type\_holiday\_id` bigint(20) NOT NULL,

PRIMARY KEY (`id`),

KEY `FK411528588A5A42F` (`type\_holiday\_id`),

CONSTRAINT `FK411528588A5A42F` FOREIGN KEY (`type\_holiday\_id`) REFERENCES `type\_holiday` (`id`)

) ENGINE=InnoDB AUTO\_INCREMENT=4 DEFAULT CHARSET=utf8;

/\*!40101 SET character\_set\_client = @saved\_cs\_client \*/;

--

-- Создание таблицы "Настройки приложения"

--

DROP TABLE IF EXISTS `config`;

/\*!40101 SET @saved\_cs\_client = @@character\_set\_client \*/;

/\*!40101 SET character\_set\_client = utf8 \*/;

CREATE TABLE `config` (

`id` bigint(20) NOT NULL,

`version` bigint(20) NOT NULL,

`last\_sw` int(11) NOT NULL,

`last\_upd` bigint(20) NOT NULL,

PRIMARY KEY (`id`)

) ENGINE=InnoDB DEFAULT CHARSET=utf8;

/\*!40101 SET character\_set\_client = @saved\_cs\_client \*/;

--

-- Создание таблицы "Типы критериев"

--

DROP TABLE IF EXISTS `type\_criterion`;

/\*!40101 SET @saved\_cs\_client = @@character\_set\_client \*/;

/\*!40101 SET character\_set\_client = utf8 \*/;

CREATE TABLE `type\_criterion` (

`id` bigint(20) NOT NULL AUTO\_INCREMENT,

`version` bigint(20) NOT NULL,

`crust` bit(1) NOT NULL,

`description` varchar(255) DEFAULT NULL,

`name` varchar(255) NOT NULL,

`type\_hours\_id` bigint(20) DEFAULT NULL,

PRIMARY KEY (`id`),

KEY `FKEE8357DCCEE3EE0F` (`type\_hours\_id`),

CONSTRAINT `FKEE8357DCCEE3EE0F` FOREIGN KEY (`type\_hours\_id`) REFERENCES `type\_hours` (`id`)

) ENGINE=InnoDB AUTO\_INCREMENT=7 DEFAULT CHARSET=utf8;

/\*!40101 SET character\_set\_client = @saved\_cs\_client \*/;

/\*!40103 SET TIME\_ZONE=@OLD\_TIME\_ZONE \*/;

/\*!40101 SET SQL\_MODE=@OLD\_SQL\_MODE \*/;

/\*!40014 SET FOREIGN\_KEY\_CHECKS=@OLD\_FOREIGN\_KEY\_CHECKS \*/;

/\*!40014 SET UNIQUE\_CHECKS=@OLD\_UNIQUE\_CHECKS \*/;

/\*!40101 SET CHARACTER\_SET\_CLIENT=@OLD\_CHARACTER\_SET\_CLIENT \*/;

/\*!40101 SET CHARACTER\_SET\_RESULTS=@OLD\_CHARACTER\_SET\_RESULTS \*/;

/\*!40101 SET COLLATION\_CONNECTION=@OLD\_COLLATION\_CONNECTION \*/;

/\*!40111 SET SQL\_NOTES=@OLD\_SQL\_NOTES \*/;

-- Dump completed on 2012-11-18 12:00:06

# Разработка приложения

Программа состоит из следующих функциональных модулей:

1. главный модуль, обеспечивающий добавление, удаление и редактирование работы сотрудников на графике, а также совместную работу всех модулей;
2. модуль ввода различных данных для создания графика и отчетности;
3. модуль импорта сотрудников из XLS-файла, обеспечивает автоматизированное управление сотрудниками;
4. модуль построения табелей по областям, где работали сотрудники;
5. модуль хранения настроек приложения;
6. модуль хранения настроек для построения отчетности;
7. модуль авторизации, обеспечивает защиту данных от неавторизованных пользователей.

Программа написана на фреймворке Grails 2.0.4, который для взаимодействия с БД использует библиотеку Hibernate 3.6.10. Hibernate позволяет абстрагироваться от БД и работать не с таблицами, а с объектами. Это в свою очередь, позволяет не привязываться к конкретной СУБД, а использовать любую, которую поддерживает Hibernate (все популярные БД и не очень, Hibernate поддерживает, например: MySQL, MS SQL, Oracle, DB 2, PostgreSQL и т.д.).

Grails использует шаблонное построение страниц для создания, просмотра, редактирования и удаления объектов. Это означает, что внешнее представление и логика работы для всех объектов одинаковая. Достаточно понять, как работать с одним объектом, чтобы начать работать со всем приложением в целом.

В приложении используется система авторизации, она построена на фильтрах. При обращении к любой странице, сначала фильтр перехватывает управление, делает проверки и только если все проверки пройдены, он выдает пользователю запрашиваемую страницу. Такой подход позволяет сосредоточить всю логику работы системы безопасности в одном месте, а не разносить ее по всем страницам. Такое удобство работы достигается использованием библиотеки Spring 3.1.0.

ГОУ ВПО

Уфимский государственный авиационный технический университет

Кафедра вычислительной математики и кибернетики

**Программный продукт**

**«Автоматизированная система построения рейтингов»**

# Руководство пользователя

**Листов 12**

2012

### Аннотация

Программный продукт (ПП) «Автоматизированная система построения графика работы на площадках добычи нефти» предназначен для построения графика работы сотрудников на площадках добычи нефти.

Кроме того, программа может применяться для построения табелей работы сотрудников по областям.

ПП может использоваться для решения как практических, так и исследовательских задач.

Данный документ описывает:

* наименование, краткую характеристику области применения программного продукта;
* основание для разработки программного продукта;
* назначение разработки, в котором указано функциональное назначение программного продукта;
* требования к программному продукту;
* состав программной документации.

### Назначение программы

Программный продукт (ПП) «Автоматизированная система построения графика работы на площадках добычи нефти» предназначен для построения графика работы сотрудников на площадках добычи нефти.

Кроме того, программа может применяться для построения табелей работы сотрудников по областям.

### Условия выполнения программы

Для выполнения программой заданных функций, ей необходимо следующее программно-аппаратной окружение:

Сервер:

1. IBM PC-совместимый компьютер с процессором – Intel® Pentium® IV или выше;
2. оперативная память – не менее 2048Mb;
3. операционная система Microsoft® Windows® XP (Vista, 7, 8), Linux, MacOS;
4. веб-сервер Apache Tomcat 7;
5. сервер баз данных MySQL (MS SQL, DB2, PostgreSQL).

Клиент:

1. IBM PC-совместимый компьютер с процессором – Intel® Pentium® III или выше;
2. оперативная память – не менее 512Mb;
3. SVGA-монитор с разрешением не ниже 1280x1024;
4. клавиатура;
5. манипулятор типа «мышь»;
6. веб-браузер Google Chrome, Opera, Firefox, IE (не ниже 10 версии);
7. операционная система Microsoft® Windows® XP (Vista, 7, 8), Linux, MacOS.

### Выполнение программы

Предполагается, что программа установлена на сервере, клиенты находятся в одной локальной сети с сервером, у клиентов установлен браузер.

Работы с программой состоит из следующих основных этапов:

1. ввод сотрудников вручную или импорт из XLS-файла;
2. ввод вспомогательных различных данных: типа дней, площадок, типа площадок, областей, подразделений, должностей сотрудников, нормы выработки за месяц, праздничных дней;
3. добавление работы сотрудников на графике;
4. Генерация табелей работы сотрудников на площадках в XLS-файл.

Для запуска программы с жесткого диска необходимо выполнить следующие действия:

1. установить веб-сервер Apache Tomcat 7;
2. установить БД MySQL и создать таблицу Tabel с пользователем tabel;
3. скопировать веб-архив с приложением в папку %TOMCAT\_HOME%/webapps;
4. запустить веб-сервер;
5. в браузере перейти по адресу: http://localhost:8080/tabel.

После запуска на экране монитора появится рабочее окно программы, в котором можно увидеть приглашение ввести имя пользователя и пароль (Рисунок 4.1).

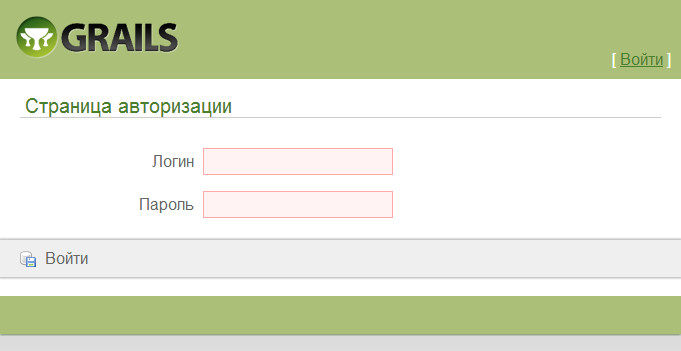


Рисунок 4.. Окно авторизации программы

После авторизации попадаем в главное окно программы, в котором можно увидеть график работы сотрудников на площадках (Рисунок 4.2).

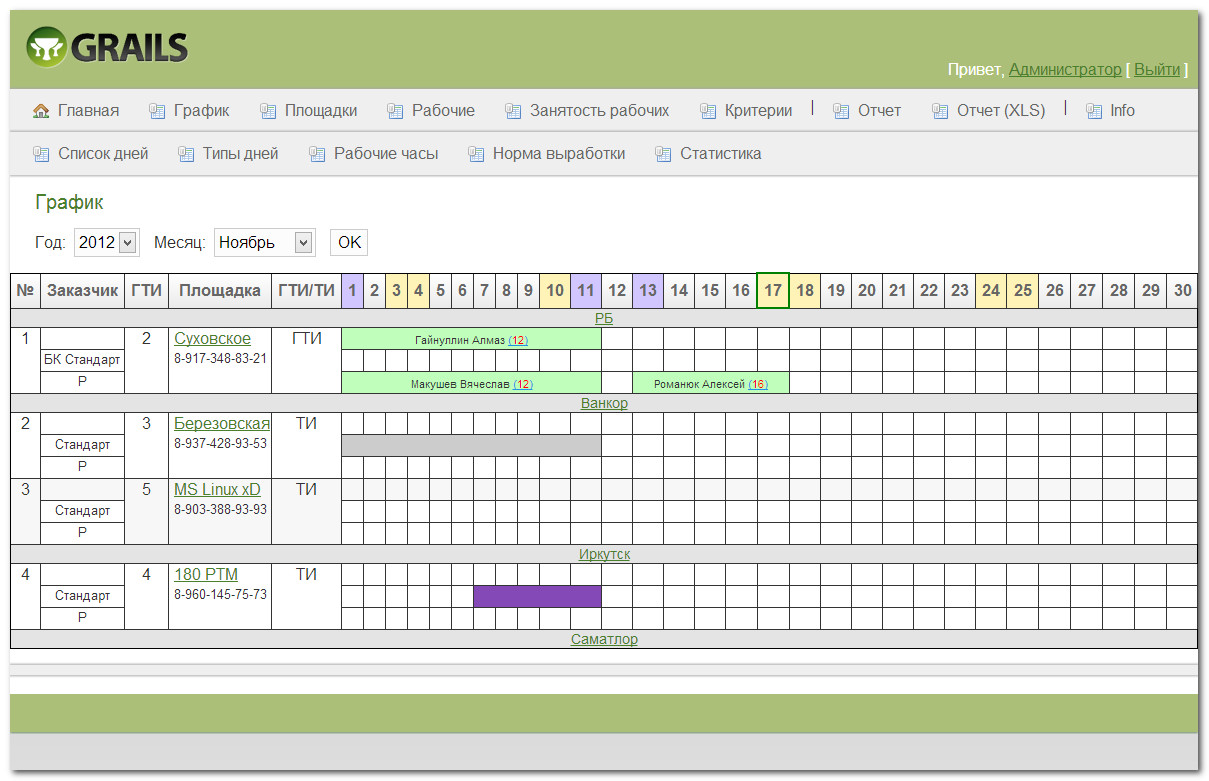


Рисунок 4.. Главное окно программы («График»)

Главное окно содержит главное меню, которое находится в верхней части окна.

Главное меню содержит пункты меню:

1. «Главная» - ссылка ведет на главное окно программы.
2. «График» - ссылка ведет на график работы.
3. «Площадки» - пункт меню, который предназначен для просмотра, добавления, редактирования и удаления площадок.
4. «Рабочие» - пункт меню, который предназначен для просмотра, добавления, редактирования и удаления рабочих.
5. «Занятость рабочих» - пункт меню, который предназначен для просмотра, добавления, редактирования и удаления занятости рабочих.
6. «Критерии» - пункт меню, который предназначен для просмотра, добавления, редактирования и удаления критериев.
7. «Отчет» - пункт меню, который предназначен для просмотра и редактирования настроек для создания отчетности.
8. «Отчет (XLS)» - генерация табелей по областям в формате XLS, с возможностью загрузки этого файла.
9. «Info» - пункт меню, доступный только администратору системы, содержит информацию о программе.

Пункт меню график содержит подпункты меню: «Список дней», «Типы дней», «Рабочие часы», «Норма выработки». Это списки, которые нужны для работы графика.

Все списки имеют одинаковую логику работы, рассмотрим работу на примере подпункта меню «Типы дней».

Нажимаем на него, открывается окно (Рисунок 4.3). Здесь выводится список всех типов дней.

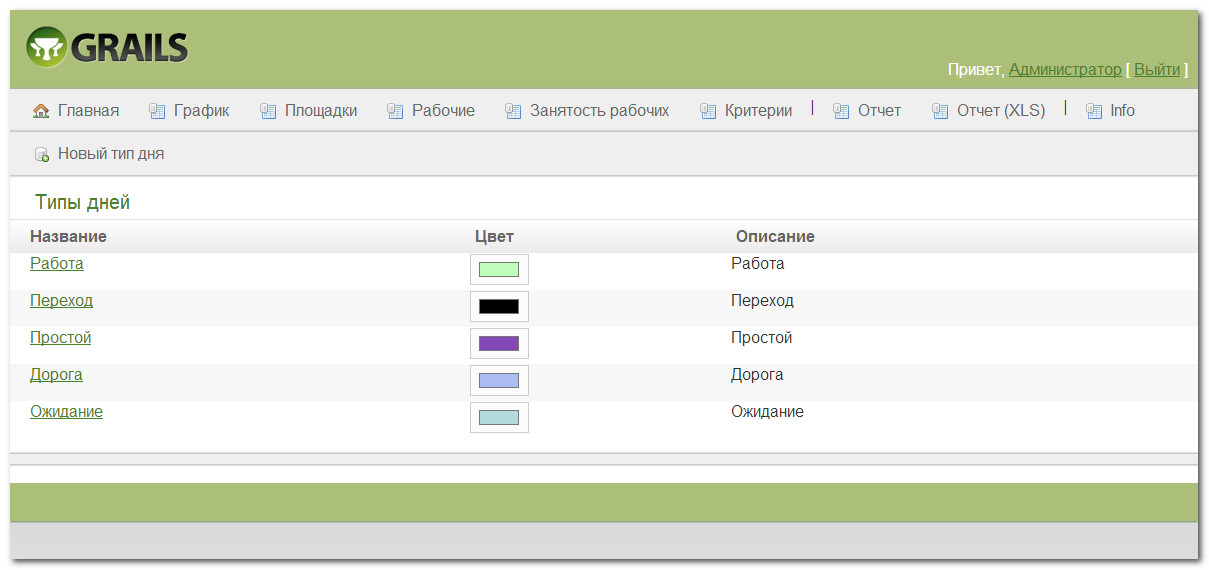


Рисунок 4.. Окно «Типа дней»

Нажатие на название списка дня, открывает окно просмотра информации о «Типе дня» (Рисунок 4.4).

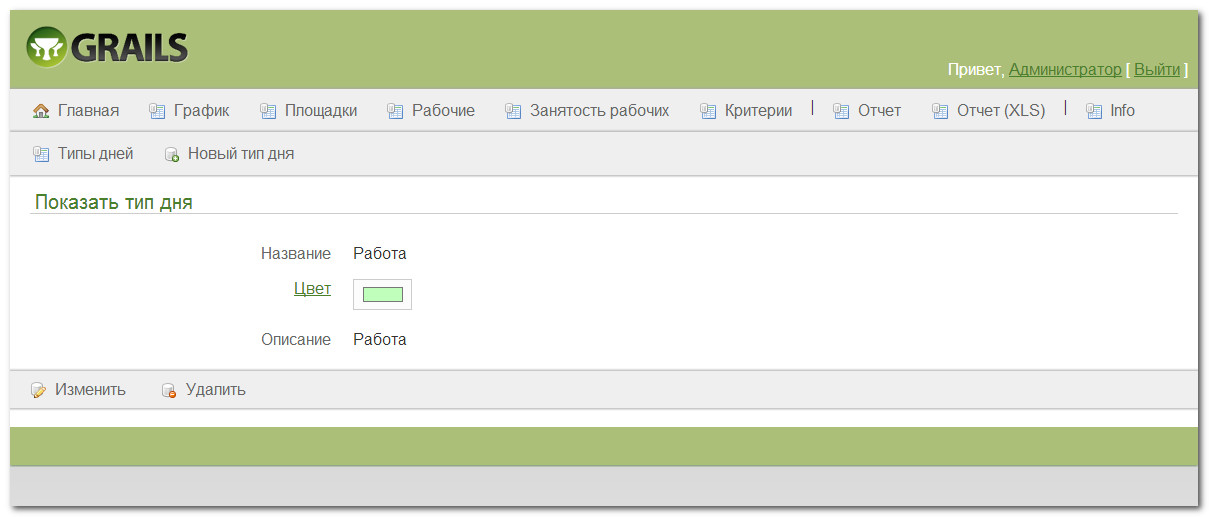


Рисунок 4.. Просмотр «Типа дня»

Этот пункт меню содержит два подпункта: «Типы дней», который ведет на список (Рисунок 4.3) и «Новый тип дня» - добавление нового «Типа дня» (Рисунок 4.5).

Под выводом всей информации «Типа дня», находятся две кнопки: редактирования информации и удаления «Типа дня». Нажатия этих кнопок приводит к соответствующим результатам (Рисунок 4.6) и (Рисунок 4.7).

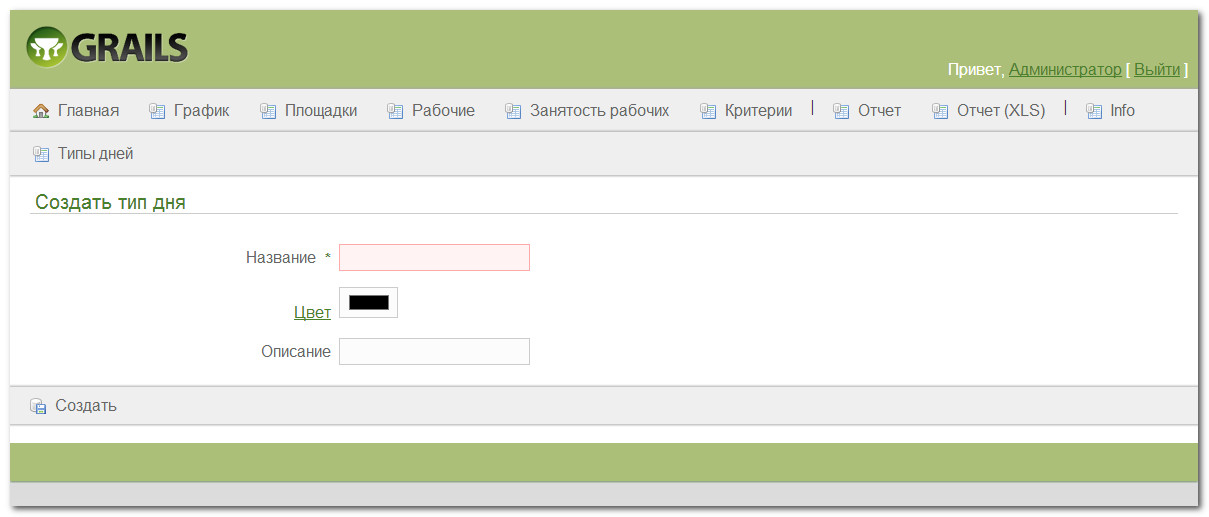


Рисунок 4.. Новый «Тип дня»

После заполнения нужные полей, обязательные поля помечены звездочкой, нажимаем кнопку «Создать». Информация добавляется в БД.

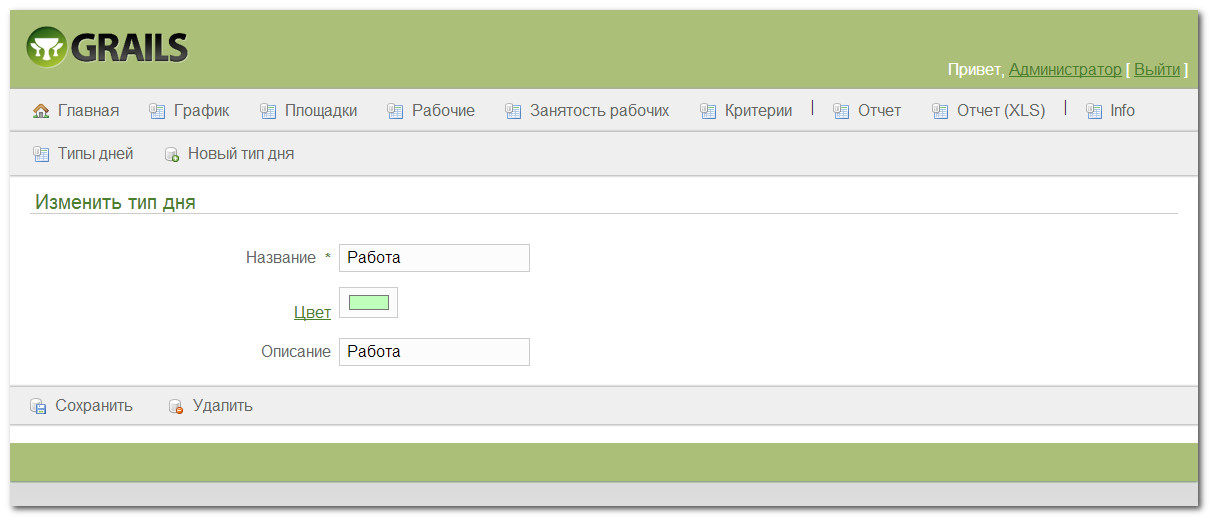


Рисунок 4.. Редактирование «Типа дня»

После модификации некоторых полей, нажимаем кнопку «Сохранить». Информация обновляется в БД.

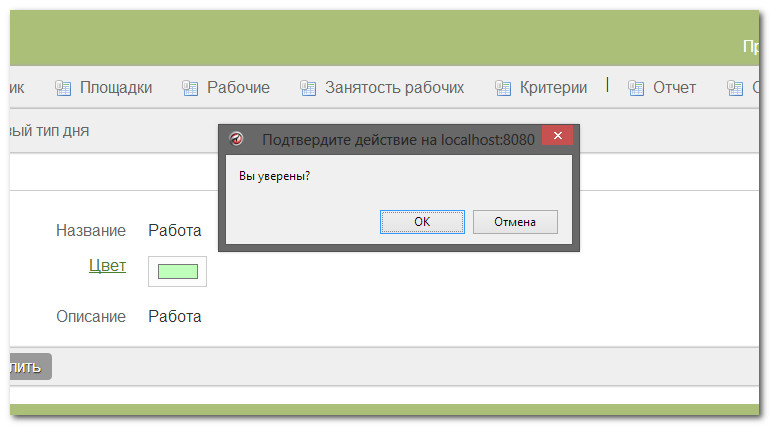


Рисунок 4.. Удаление «Типа дня»

При нажатии на кнопку «ОК», информация о «Типе дня» удаляется из БД.

C остальными справочниками работа происходит точно так же.

#### Добавление работы на график

Для добавления работы на график, сначала нужно выбрать интересующую нас площадку, потом интересующую нас строку, затем день и нажать на нужную клетку. В результате откроется окно (Рисунок 4.8).

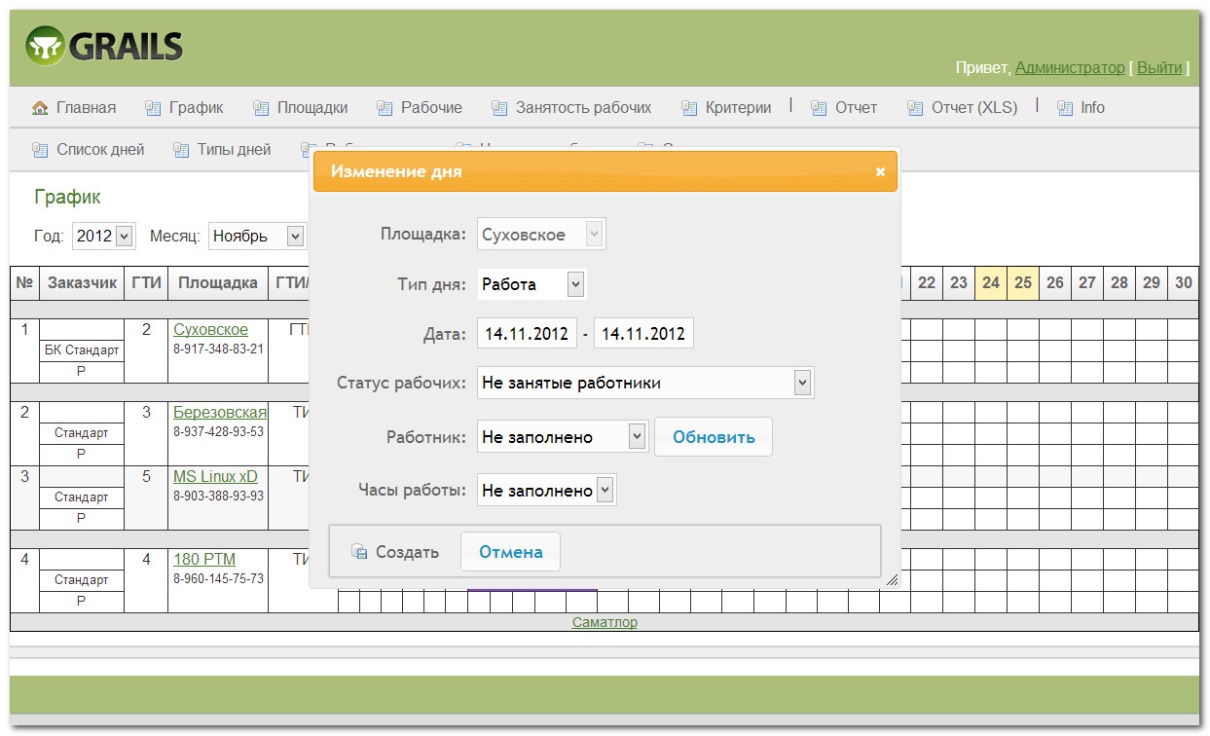


Рисунок 4.. Добавление работы на график

Выбираем тип дня из выпадающего списка, дата начала будет соответствовать выбранному нами дню, дату конца нужно выбрать с помощью календаря, который открывается по нажатию на дату.

Выбираем статус рабочего из выпадающего списка, после этого происходит Ajax-запрос и в выпадающий список сотрудников, попадают только те, которые соответствую выбранному статусу, на выбранный диапазон дней.

Выбираем часы работы и нажимаем кнопку «Создать», работа добавлена на график (Рисунок 4.9).

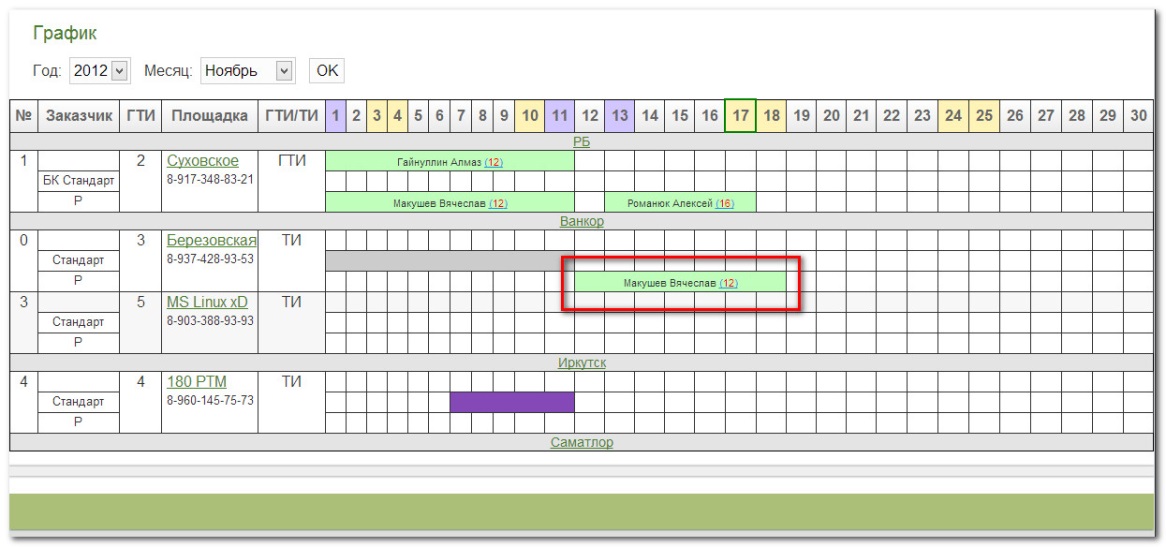


Рисунок 4.. Добавленная работа на графике

Для редактирования работы, нужно щелкнуть на добавленную работу, откроется окно (Рисунок 4.10).

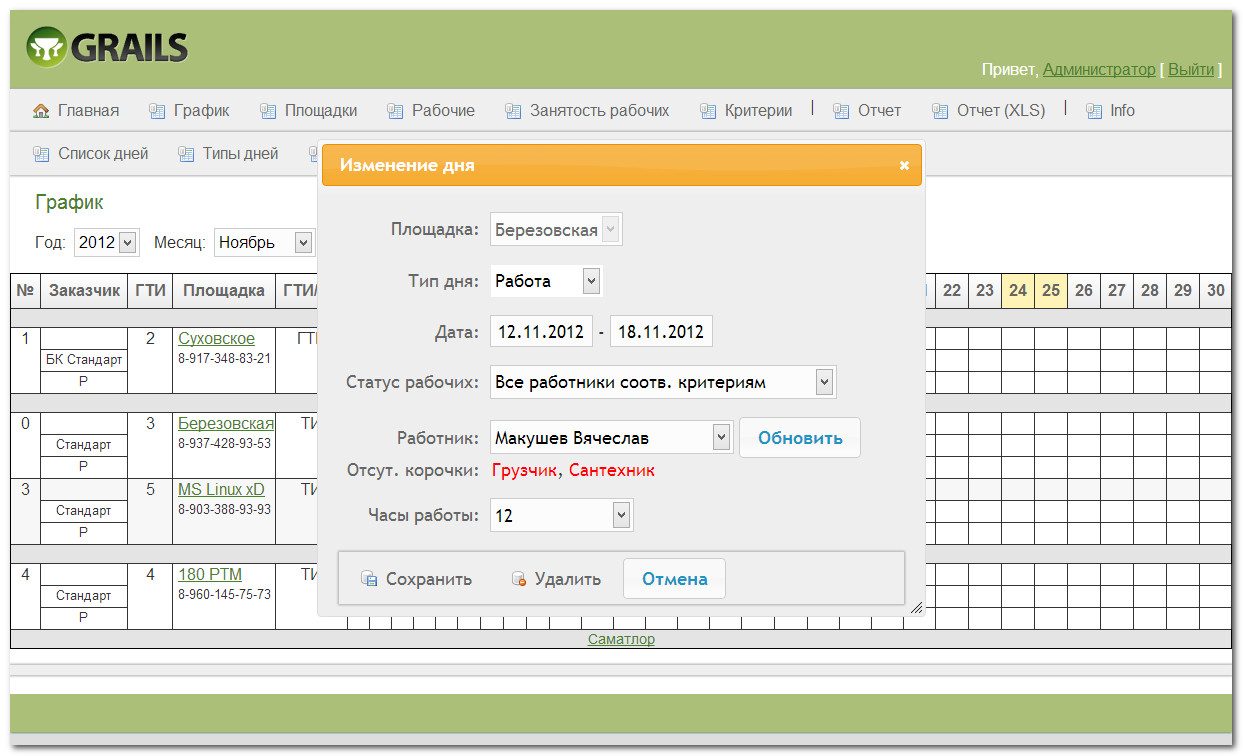


Рисунок 4.. Изменение работы на графике

После модификации некоторых полей, нажимаем кнопку «Сохранить». Информация обновляется в БД и на графике.

Для удаления работы, нужно нажать на кнопку «Удалить», выскочит окно с подтверждением на которое следует ответить «ОК».

#### Статистика работы по графику

Для просмотра статистики работы по графику нужно на странице графика (Рисунок 4.2), выбрать пункт меню «Статистика», выведется статистика (Рисунок 4.11).

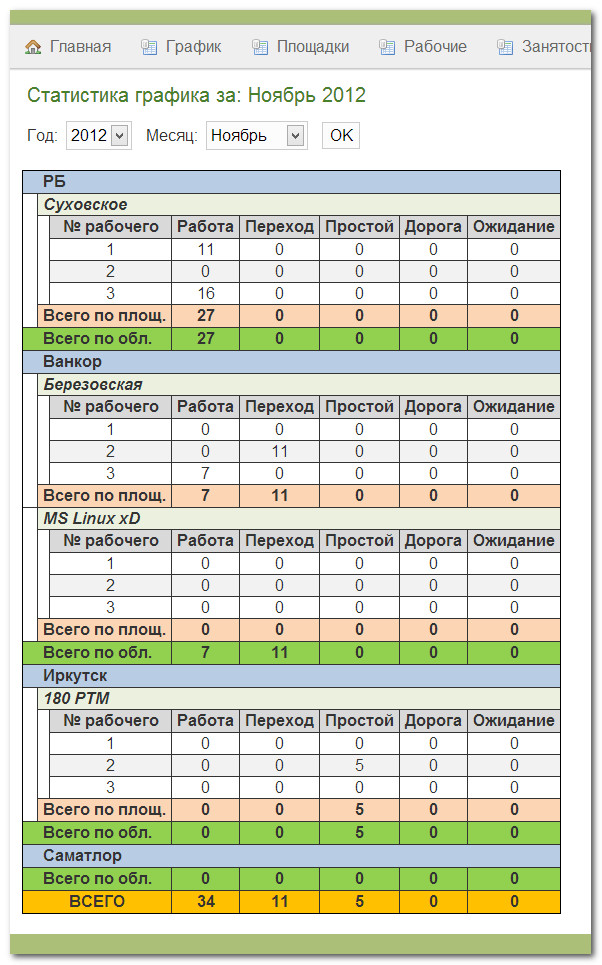


Рисунок 4.. Статистика работы по графику

#### Генерация табелей по областям

Для генерации табелей по областям в формате XLS, нужно на главном меню нажать на кнопку «Отчет (XLS)». Произойдет генерация отчета, а потом браузер предложит вам сохранить полученный файл на жестком диске (Рисунок 4.12).

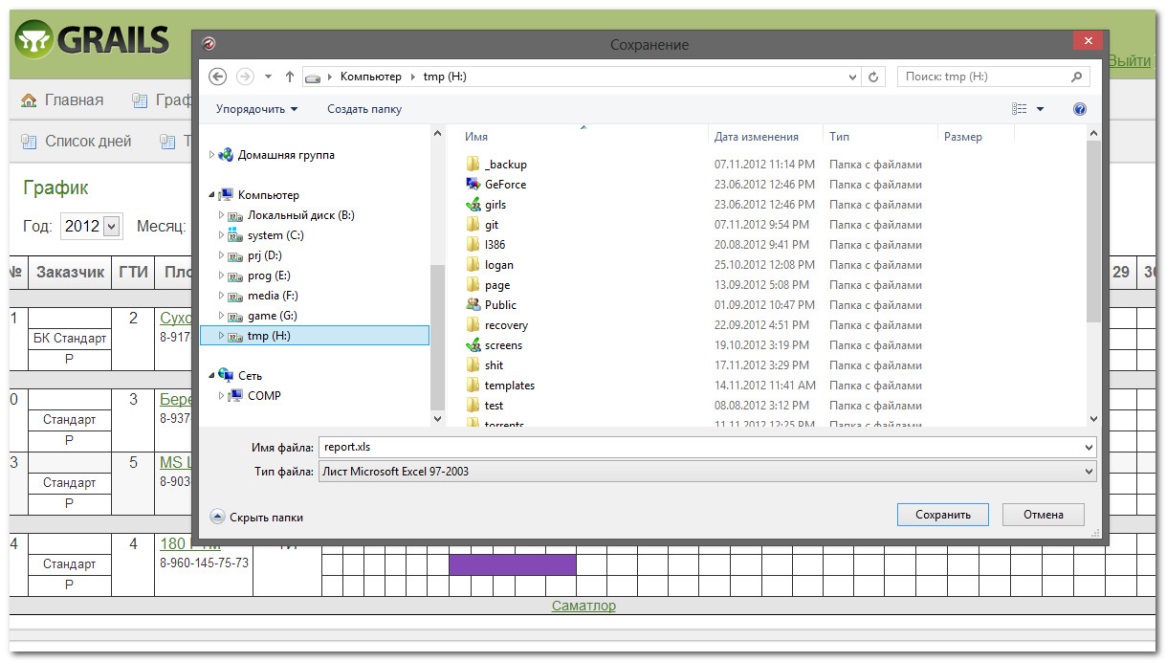


Рисунок 4.. Сохранение отчета на жестком диске

Скачанный отчет можно открыть в MS Excel и он будет иметь вид (Рисунок 4.13).

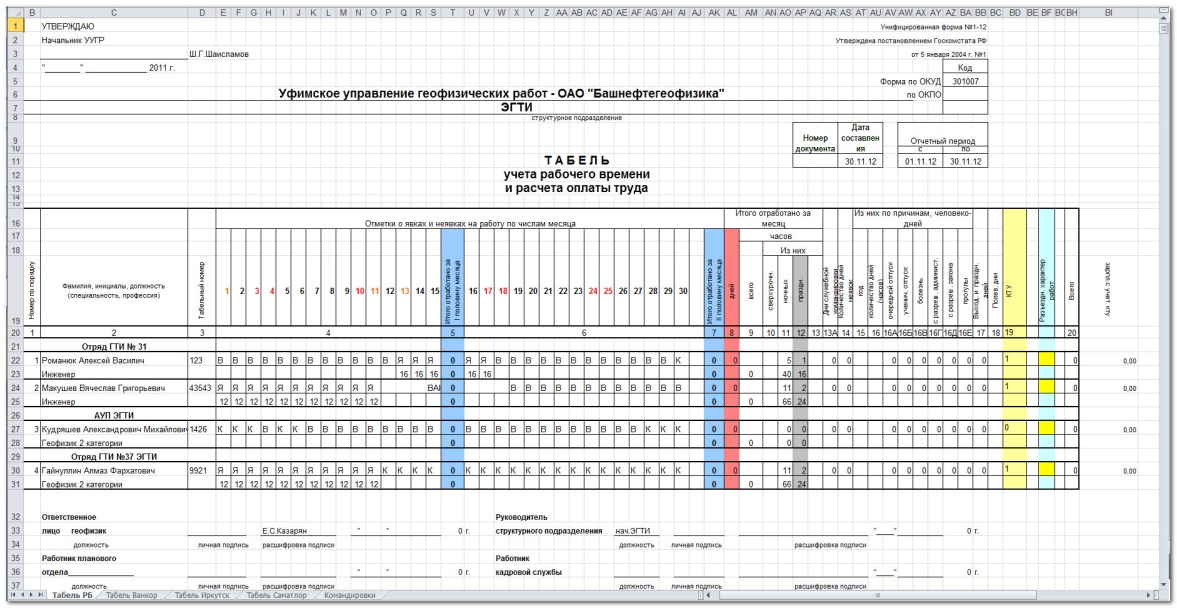


Рисунок 4.. Отчет в формате XLS

### Сообщения оператору

В программе предусмотрены сообщения оператору в случае возникновения ошибки. Сообщения оператору, их причина, а также необходимые действия для устранения причины приведены в таблице 4.1.

Таблица 4.1

Сообщения оператору

|  |  |  |
| --- | --- | --- |
| Сообщение | Причина | Действия |
| Ошибка: Дата конца < даты начала. | Пользователь выбрал дату конца работы меньшую даты начала. | Выбрать дату конца больше чем дату начала. |
| Ошибка: Не все поля заполнены | При добавлении работы, пользователь заполнил не все поля в красной рамке. | Заполнить все поля отмеченные красной рамкой. |
| Рабочий не найден с id: 10 | Пользователь пытается получить информацию о несуществующем рабочем по ссылке. | Найти нужного рабочего через пункт меню «Рабочие». |
| Заполните это поле. | При попытке сохранить информацию с незаполненными полями. | Заполнить поля помеченный звездочкой и красной рамкой. |
| Ошибка чтения файла | Пользователь пытается импортировать файл с неправильной структурой. | Импортировать файл снова с правильной структурой. |
| Доступ запрещен. | Пользователь пытается обратиться по ссылке на недоступную для него страницу. | Обратиться к администратору систему для изменения роли пользователя. |
| Неправильный пароль | Пользователь ввел не правильный пароль для своего логина. | Ввести правильный пароль или обратиться к администратору системы за восстановлением пароля. |
| Пользователь с таким логином не найден | В БД нет пользователя с введенным логином. | Ввести существующий логин или обратиться к администратору системы за регистрацией пользователя. |

# Заключение

В соответствии с поставленной задачей в курсовой работе было выполнено следующее:

1. Изучена работа планирования работы в ЭГТИ;
2. разобран алгоритм построения графика работы на площадках;
3. спроектирована база данных;
4. разработана программа на Java, реализующая данный алгоритм.