


Министерство образования и науки Российской Федерации
Федеральное государственное бюджетное образовательное учреждение высшего образования

АМУРСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
(ФГОУ ВО «АмГУ»)

Факультет энергетический
Кафедра автоматизации производственных процессов и электротехники
Направление подготовки 15.03.04 – Автоматизация технологических процессов и производств
Профиль Автоматизация технологических процессов и производств в энергетике

ДОПУСТИТЬ К ЗАЩИТЕ

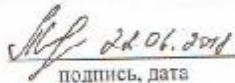
И.о. зав. кафедрой

 О.В. Скрипко
« 28 » июня 2018г.

БАКАЛАВРСКАЯ РАБОТА

на тему: Создание автоматизированной системы планирования и учета работ по АСУ ТП на Благовещенской ТЭЦ

Исполнитель
студент группы 441- об.


подпись, дата

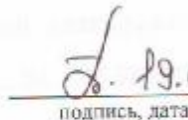
Л.С. Афанасов

Руководитель
доцент, канд. техн. наук.


подпись, дата

Д.А. Теличенко

Консультант по безопасности
и экологичности
доцент, канд. техн. наук


подпись, дата

А.Б. Булгаков

Нормоконтроль
профессор, д-р техн. наук.


подпись, дата

О.В. Скрипко

Благовещенск 2018

Министерство образования и науки Российской Федерации
Федеральное государственное бюджетное образовательное учреждение высшего
профессионального образования


АМУРСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
(ФГБОУ ВО «АмГУ»)

Факультет энергетический

Кафедра автоматизации производственных процессов и электротехники

УТВЕРЖДАЮ

И. о. зав. кафедрой

 О. В. Скрипко
подпись

« 28 » июня 2018г.

ЗАДАНИЕ

К выпускной квалификационной работе студента 441 гр. Афанасова Леонида Сергеевича

1. Тема выпускной квалификационной работы: Создание автоматизированной системы планирования и учета работ по АСУ ТП на Благовещенской ТЭЦ (комплексная выпускная квалификационная работа)

(утверждено приказом от 09.04.2018, №772-уч)

2. Срок сдачи студентом законченной работы: 30 июня 2018г

3. Исходные данные к выпускной квалификационной работе: отчет по практике; имеющиеся наработки на предприятии по ведению учета и планирования; учебный план направления подготовки 15.03.04. «Автоматизация технологических процессов и производств»

4. Содержание выпускной квалификационной работы:

- 1) Описание объекта автоматизации
- 2) Концептуальное проектирование системы
- 3) Разработка структуры базы данных

4) Разработка структуры пользовательского интерфейса

5) Программная реализация

6) Руководство пользователя

7) Безопасность жизнедеятельности

5. Перечень материалов приложения:

Лист 1: Исходная постановка задачи

Лист 2: Структурная схема информационной системы

Лист 3: Структура интерфейса пользователя

Лист 4: Алгоритмическая схема информационное системы

Лист 5: Программная реализация информационной системы

Лист 6: Руководство пользователя

6. Дата выдачи задания: 23 января 2018

Руководитель выпускной квалификационной работы: Теличенко Денис Алексеевич, доцент кафедры АПП и Э, к.т.н.

Задание принял к исполнению (дата): 23 января 2018



РЕФЕРАТ

Выпускная квалификационная работа содержит 120 с., 58 рисунков, 9 таблиц, 5 приложений.

ОБЪЕКТ АВТОМАТИЗАЦИИ, ЭСКИЗНОЕ ПРОЕКТИРОВАНИЕ, ИНТЕРФЕЙС ПОЛЬЗОВАТЕЛЯ, СУБД, РАЗРАБОТКА СТРУКТУРЫ, ПЛАНИРОВАНИЕ И УЧЕТ, ПРЕДВАРИТЕЛЬНОЕ ПРОЕКТИРОВАНИЕ, АВТОМАТИЗАЦИЯ ПЛАНИРОВАНИЯ И УЧЕТА РАБОТ, ПОДСИСТЕМА, ИНФОРМАЦИОННАЯ СИСТЕМА

В выпускной квалификационной работе было задание автоматизировать процесс планирования и учета работ ремонтного персонала по АСУ ТП на Благовещенской ТЭЦ.

В ходе выполнения выпускной квалификационной работы был приведен принцип планирования ремонтных работ в настоящее время на Благовещенской ТЭЦ и была:

- разработана структурная схема информационной системы;
- спроектирована структура пользовательского интерфейса ИС;
- спроектирована структура базы данных ИС;
- разработана алгоритмическая схема ИС;
- приведена программная реализация системы.

СОДЕРЖАНИЕ

Определения, обозначения, сокращения	8
Введение	9
1 Объект автоматизации	10
1.1 Общие сведения о предприятии	10
1.2 Общее описание существующего решения задачи.....	12
1.3 Основные недостатки существующего решения.....	16
1.4 Техническое задание на разработку	17
2 Концептуальное проектирование системы.....	18
2.1 Обоснование выбора программного обеспечения для реализации информационной системы.....	19
2.2 Предварительное проектирование информационной системы	22
2.2.1 Работа руководителя с информационной системой	24
2.2.2 Работа ремонтного персонала с информационной системой ..	25
2.2.3 Работа оперативного персонала с системой.....	25
2.2.4 Работа администратора с информационной системой.....	26
2.3 Подсистема добавления нового пользователя	26
2.4 Подсистема авторизации пользователя	27
2.5 Подсистема планирования	27
2.6 Подсистема составления отчетности	28
2.7 Подсистема фильтрации данных	28
3 Разработка структуры базы данных	29
3.1 Атрибуты сущности «Пользователи»	29
3.2 Атрибуты сущности «Дата планирования».....	31

3.3 Атрибуты сущности «Задачи»	32
4 Разработка структуры интерфейса пользователя	37
4.1 Главная страница.....	37
4.2 Форма регистрации пользователя	40
4.3 Страница добавления плана.....	41
4.4 Страница просмотра списка сотрудников	42
4.5 Страница просмотра спланированных задач	42
4.6 Страница зарегистрированных пользователей	44
5 Программная реализация.....	45
5.1 Выбор языка программирования.....	46
5.2 Реализация подсистем информационной системы.....	47
5.2.1 Реализация подсистемы авторизации	47
5.2.2 Реализация подсистемы регистрации	49
5.2.3 Реализация подсистемы планирования ремонтных работ.....	51
5.2.4 Реализация подсистемы сохранения данных для отчетности .	53
5.3 Описание запросов к сущностям базы данных	55
6 Руководство пользователя информационной системы	58
6.1 Авторизация пользователя.....	58
6.2 Регистрация нового пользователя	59
6.3 Планирование ремонтных работ	60
6.4 Фильтрация данных в таблице.....	65
6.5 Сохранения файла для отчетности	66
6.6 Подтверждение спланированных задач.....	67
7 Безопасность жизнедеятельности.....	69
7.1 Безопасность	69

7.2 Экологичность	71
7.3 Чрезвычайные ситуации	72
Заключение	74
Библиографический список	75
Приложение А – Техническое задание	77
Приложение Б–Главная страница.....	86
Приложение В – Страница планирования задач.....	87
Приложение Г – Страница списка сотрудников	88
Приложение Д – Листинг программы.....	89

ОПРЕДЕЛЕНИЯ, ОБОЗНАЧЕНИЯ, СОКРАЩЕНИЯ

АСУ ТП –автоматизированная систему управления технологическим процессом;

БТЭЦ – Благовещенская теплоэлектроцентраль;

ПО – программное обеспечение;

ИС – информационная система;

МНУА – монтажно-наладочный участок автоматики;

СУБД – система управления базами данными;

ПЭВМ – персональная электронно-вычислительная машина;

ВДТ – видеодисплейный терминал.

ВВЕДЕНИЕ

Планирование деятельности организации предполагает разработку комплекса мероприятий, которые определяют последовательность достижения целей, учитывая возможности для наиболее эффективного использования ресурсов. Организации, которые планируют свою деятельность, работают более успешно, чем организации, которые не планируют свою деятельность.

Планирование и учет ремонтных работ является обязательной составляющей для ТЭЦ. Так как на предприятии имеется много оборудования, которое в свою очередь необходимо обслуживать, может выйти из строя, или появляется необходимость замены на более современное. В связи с этим проводятся организационные работы по отладке, ремонту, поддержания оборудования или обновление ПО. По всем проделанным работам необходимо вести отчетность о выполненных или невыполненных работах.

В настоящее время данный процесс планирования и отчета не имеет единой базы, а также нет возможности одновременной работы с файлом нескольких сотрудников, что значительно замедляет процесс. Автоматизировать данный процесс будет актуально и заметно сократит время, затрачиваемое на данное мероприятие.

Для реализации автоматизации планирования и учета работ ремонтного персонала разрабатывается информационная система, которая осуществляет запросы к базе данных (единому архиву на сервере) и вывода информации при помощи интерфейса пользователя. А также сохранение информации в excelфайл.

1 ОБЪЕКТ АВТОМАТИЗАЦИИ

В качестве объекта автоматизации рассматривается задача планирования и учета ремонтных работ рабочего персонала на Благовещенской ТЭЦ.

Технологический процесс на любом предприятии обусловлен наличием оборудования, инструментов, обслуживающего персонала и т.д. Любое техническое средство имеет срок эксплуатации, возможны незначительные поломки, а также аварийный износ. При поломке оборудования производят восстановления, а также при поступлении современного оборудования в оборот (нового подхода к решению задач) осуществляют замену старого на новое. Исходя из всего этого следует, что, учет и планирование ремонтных работ является неотъемлемой частью для любого предприятия.

Прежде чем переходить к описанию процедуры планирования и учета ремонтных работ, рассмотрим общую структуру предприятия БТЭЦ и задачи монтажно-наладочного участка автоматики, как один из типичных участков, на котором требуется проводить планирование и учет различных работ.

1.1 Общие сведения о предприятии

Благовещенская ТЭЦ - крупнейшее промышленное предприятие города Благовещенска. ТЭЦ предназначена для централизованного теплоснабжения.

Благовещенская ТЭЦ является паротурбинной электростанцией. В качестве основного топлива на ТЭЦ используется Райчихинский бурый уголь, и в качестве резервного топлива используется топочный мазут марки М-100, который хранится в 4-х металлических баках емкостью 2000 м³.

Установленная электрическая мощность станции 404 МВт, теплофикационная - 1005 Гкал/ч.[1]

Монтажно-наладочный участок автоматики (МНУА) входит в состав структурного подразделения Благовещенская ТЭЦ (СП БТЭЦ) филиала «Амурская генерация» АО «ДГК». В состав МНУА входит ремонтный и эксплуатационный персонал, который в свою очередь разделен на:

- группа инженеров по наладке и испытаниям;

- электромонтажная группа;
- группа по ремонту и обслуживанию АСУ ТП.

Структура МНУА представлена на рис. 1.



Рисунок 1 -Структура МНУА

Основные задачи участка:

- выполнение проектных, монтажных, ремонтных, наладочных работ в срок и с высоким качеством;
- разработка, внедрение, обслуживание программного обеспечения АСУ ТП;
- повышение производительности труда;
- содержание закрепленного за участком оборудования, зданий и сооружения в состоянии эксплуатационной готовности;
- соблюдение требований промышленной и пожарной безопасности в процессе ремонта и эксплуатации оборудования;
- выполнение требований охраны труда.

Основные функции МНУА:

- проектирование, монтаж, ремонт и наладку релейной защиты, КИП и А и электрооборудования;
- разработка, внедрение в производство и обслуживание программного обеспечения АСУ ТП;
- расследование причин аварий и отказов оборудования;
- разработка мероприятий по повышению надежности и экономичности работы оборудования, повышению производительности труда;

- внедрение новых методов контроля, в совершенствовании схем управления, защит и автоматики;
- совместно с цехами электростанции принимает участие во внедрении утвержденных мероприятий по науке и новой технике;
- организует сдачу в эксплуатацию из монтажа, наладки и ремонта оборудования;
- содержание в исправном состоянии средства пожаротушения на участке;
- обеспечивает рабочие места необходимым инвентарем, приборами, инструментами и документацией;
- проводит мероприятия по повышению квалификации персонала участка;
- организует учебу и составляет отчетность о производственной деятельности, ведет техническую документацию;
- содержит в чистоте помещения и территорию, закрепленную за участком.

В основные задачи МНАУ входит: наладка и ремонт оборудования, поддержание оборудования в исправности и составления отчетности – рассмотрим подробнее данную задачу.

1.2 Общее описание существующего решения задачи

Каждый день вначале рабочего дня (с 8:00 до 9:00 рабочего времени) осуществляется планирование задач на весь день для обслуживающего персонала. Составлять план может как руководитель, так и исполнитель (планировать задачи для себя). Задачи на исполнение берутся из журнала дефектов, либо те, которые не были выполнены по какой-либо причине ранее. Шаблон заполнения таблицы планирования, который ведется для группы по ремонту и обслуживанию АСУТП, представлен на рис. 2.

После составления плана сохраненный файл отправляют на сервер, доступ к которому имеют как руководитель, так и исполнитель.

Заполняется исполнителем/руководителем (План)						
1	Дата	День недели	ФИО	Время, факт	Задача	Описание
2	26.03.2018	Пн		08.00-09.00	Планерка, анализ дефектов, оценка эффективности прошедшей работы.	
3				09.00-10.00		
4				10.00-11.00		
5				11.00-12.00		
6				13.00-14.00		
7				14.00-15.00		
8				15.00-16.00		
9				16.00-17.00	Выполнение журналов, планы на сб/б, устранение дефектов, планирование	
10				внеплан.		
11				офиц.рем.		
12				*		
13				учет часов		
14						

Рисунок 2 – Шаблон таблицы составления плана

В представленной выше части таблицы (рис. 2) имеются столбцы для заполнения:

- дата (тип данных данного поля является «дата», формат записи значения – «дд.мм.гг»);
- день недели (значение в поле вставляется автоматически, при записи значения в поле «дата»);
- ФИО (тип данных поля – текст, значение поля выбирается из выпадающего списка, в которое занесены исполнители);
- время(формат поля является текстовым, здесь указывается значение времени в течение которого выполняется определенная задача, также присутствуют строки «внеплан.», «офиц.рем.» и «*»);
- задача (тип данных данного поля является текст, в поле описывается задание на выполнение);
- описание (тип данных – текст, заполняется описание конкретной задачи).

В столбце фактическое время присутствуют такие строки, как:

- «внеплан.» (заполняется в том случае, если произошла нештатная ситуация, и исполнитель устранял данную проблему, не зависимо от запланированных задач);

- «офиц. рем.»;

- «*» (используется для удобства руководителя, заполняет невыполненные задачи, которые были запланированы ранее).

По завершению каждой задачи исполнитель составляет отчетность (рис. 3), выполнено или не выполнено. Если задача была не выполнена по какой-либо причине, исполнитель обязан написать обоснование.

В конце рабочего дня (с 16:00-17:00) проходит совещание, где исполнитель предоставляет полный отчет выполненных работ за день, а руководитель в свою очередь заполняет примечание(рис. 4) по проделанной работе. Руководитель может результат проделанной работы подтвердить (выполненная работа полностью соответствует указанному заданию), не подтвердить (выполнена не в соответствии с указанным заданием, либо совсем не выполнена), перенести (если исполнитель не успел выполнить задачу из-за нехватки времени) или снять(из-за неактуальности или передачи задания другому исполнителю).

Заполняется исполнителем (Факт)	
Результат	Обоснование
Выполнено	
Не выполнено	пример обоснования

Рисунок 3 – Шаблон отчетности работы исполнителем

Заполняется руководителем
Примечание <input type="text"/>
Подтверждено
Не подтверждено
Перенесено
Снято

Рисунок 4 – Факт выполненной работы

Как видно из рис.3 и рис.4заполняются еще три столбца:

- результат (тип поля является текст, исполнитель устанавливает факт о проделанной работе, выбор возможен из двух значений «выполнено» и «не выполнено»);

- обоснование (формат поля – текст, исполнитель пишет обоснование о не выполненной задаче);

- примечание (тип данного поля текст, руководитель выбирает значение из выпадающего списка, возможен выбор – «подтверждено», «не подтверждено», «перенесено» и «снято»).

На рис. 5 приведены еще столбцы, которые не были рассмотрены выше, такие как:

- допуск (формат поля является текст, заносимые значения выбираются из выпадающего списка. В списке выводятся ФИО сотрудников);

- учет ремонта (формат поля – текст, можно выставить только два значения – это «да» или «нет», выбор происходит из выпадающего списка);

В случае с информационной системой, обеспечивается корректная работа нескольких пользователей одновременно, что значительно ускоряет процесс планирования и учета исполнителей.

Благодаря разделению прав доступа к изменению данным (планирование и учет), исключается возможность исполнителю выставить примечание задачи «подтверждено» вместо руководителя, менять данные других сотрудников, ставить допуск к работе без прохождения инструктажа и тому подобное.

ИС позволяет, не полностью, но большую часть процесса планирования автоматизировать, благодаря «подтягиванию» задач, которые не были выполнены ранее исполнителем или из файла, к примеру «Журнала дефектов».

1.4 Техническое задание на разработку

Техническое задание для выпускной квалификационной работы разработано согласно требованиям ГОСТ 34.602-89 «Техническое задание на создание автоматизированной системы». Техническое задание приведено в приложении А.

2 КОНЦЕПТУАЛЬНОЕ ПРОЕКТИРОВАНИЕ СИСТЕМЫ

Проектирование автоматизированной системы планирования и учета рабочего персонала охватывает три основные области:

1) проектирование объектов данных, которые будут реализованы в базе данных;

2) проектирование программ, экранных форм, отчетов, которые будут обеспечивать выполнение запросов к данным;

3) учёт конкретной среды или технологии, а именно: топологии сети, конфигурации аппаратных средств, используемой архитектуры (файл-сервер или клиент-сервер), параллельной обработки, распределённой обработки данных и т.п.

Системное проектирование включает в себя следующие стадии:

- определение общих целей проектирования с формированием локальных (отдельных) целей разработки;

- формирование концепции системы (объекта исследования) и подготовки данных для создания модели объекта;

- разработки описания системы в виде структур объекта проектирования и построения функциональных подсистем объекта;

- формализация задач проектирования, в том числе формирование области поиска решений, систем предпочтений и ограничений, требований к объекту и т.п.

Основные этапы концептуального проектирования являются:

- предварительное проектирование;

- эскизное проектирование;

- изготовление, испытания и доводка опытного образца системы.

2.1 Обоснование выбора программного обеспечения для реализации информационной системы

Для проектирования ИС необходимо определиться с выбором инструментов разработки. Для реализации интерфейса пользователя существуют такие ПО как:

1) Balsamiq Mockups[2] – это сервис для построения макетов пользовательских интерфейсов веб-сайтов, мобильных приложений и обычных программ. Balsamiq доступен для всех операционных систем. Также есть онлайн-версия Balsamiq. Лицензия для одного ПК обойдется в 89 долларов, а доступ к веб-приложению — от 9 долларов в месяц (не более двух активных проектов).

2) Wireframe Sketcher Studio[3]- поставляется как автономный макет и каркасный инструмент, а также в виде плагина для Eclipse IDE. Данное ПО является кроссплатформенным и доступен на Windows, Mac OS X и Linux. Встроенный рефакторинг позволяет «безболезненно» переименовывать и перемещать файлы без разрыва связей. Гибкий пользовательский интерфейс позволяет использовать преимущества нескольких мониторов. Бессрочная лицензия, включающая в себя один год технической поддержки одного пользователя, обойдется в 99 долларов.

3) 1С ERPУправление предприятием 2[4] является наилучшим инструментом для создания комплексной системы управления работой организаций с разноплановой деятельностью. Также при создании функционала прикладного решения 1С ERP были учтены требования крупных компаний, что позволило сделать программу более расширенной и востребованной. Среди главных особенностей: поддержка работы через Интернет, в режиме «облачных» технологий, а также через мобильные устройства на операционных системах iOS и AndroidOS. Стоимость ПО составляет 360 тыс. руб.

4) Написание собственного WEB- приложения с необходимым функциями для решения поставленной задачи. В настоящее время большое

распространение получили WEB -Приложения. WEB – приложения позволяют создавать сайты различной сложности и структуры для решения практически любых задач.

Проектирование собственного WEB-приложения будет разумнее и эффективнее, из-за:

- значительного снижения денежных затрат;
- того, что не будет задействован максимальный функционал приведенного выше ПО;
- присутствия гибкости системы под конкретные поставленные задачи.

А для хранения всех данных нужно определиться с выбором СУБД, для хранения данных в сетевом режиме существует три вида:

- 1) SQLite[5] – очень мощная встраиваемая система управления.
- 2) MySQL[6] – самая популярная и распространенная СУБД.
- 3) PostgreSQL[7] – наиболее продвинутая СУБД.

В таблице 1 приведены преимущества и недостатки данных СУБД.

Таблица 1 – Преимущества и недостатки СУБД

Наименование СУБД	Недостатки	Преимущества
1	2	3
SQLite	- отсутствие системы пользователей (более крупные СУБД включают в свой состав системы управления правами доступа); - отсутствие возможности увеличения производительности;	- вся БД состоит из единого файла, что позволяет легко переносить на разные машины; - может показаться что эта СУБД примитивна, но она использует SQL;

1	2	3
	<p>- нет возможности записи больших объемов данных.</p>	<p>- отличная при разработке и тестировании, в процессе разработки приложений часто появляется масштабирования.</p>
<p>MySQL</p>	<p>- из-за некоторых способов обработки данных MySQL (связи, транзакции, аудиты) иногда уступает другим СУБД по надежности; - медленная разработка, хотя MySQL технически открытое ПО, существуют жалобы на процесс разработки; - по задумке в MySQL заложены некоторые ограничения функционала, которые иногда необходимы в особо требовательных приложениях.</p>	<p>- простота в работе, установить MySQL довольно просто; - богатый функционал, MySQL поддерживает большинство функционала SQL; - большое количество функций, обеспечивающих безопасность; - MySQL легко работает с большими объемами данных и легко масштабируется.</p>
<p>PostgreSQL</p>	<p>- при простых операциях чтения PostgreSQL может значительно замедлить сервер и быть медленнее своих конкурентов, таких как MySQL;</p>	<p>- бесплатное ПО с открытым исходным кодом; - существует довольно большое сообщество, в котором вы запросто</p>

1	2	3
	- по своей природе, популярностью эта СУБД похвастаться не может, хотя и присутствует довольно большое сообщество.	найдёте ответы на свои вопросы; - существует возможность расширения функционала за счет сохранения своих процедур.

Для хранения данных была выбрана СУБД MySQL, поскольку она является простой в использовании, доступной и самая популярная среди приведенных выше СУБД.

Следующим шагом рассмотрим предварительное проектирование ИС. Система состоит из следующих подсистем:

- добавление пользователя;
- авторизация;
- планирование;
- составление отчетности;
- просмотр эффективности работы сотрудников;
- фильтрация данных.

2.2 Предварительное проектирование информационной системы

Работа с представленной ИС проводится лицами разной рабочей категории. В связи с этим будет разумно разделить пользователей на разные группы со своими правами доступа. Таких разделений будет четыре: руководитель, ремонтный, оперативный персонал и администратор, который будет подтверждать регистрацию в ИС и вносить какие-либо необходимые изменения. А также пока пользователь не авторизуется он будет находиться в гостевом режиме. В данном режиме никакой функционал не будет доступен для пользователя, кроме авторизации и регистрации.

В таблице 2 представлены права доступа групп пользователей.

Таблица 2 – Права доступа пользователей

Наименование	Функционал
Руководитель	<ul style="list-style-type: none">- список сотрудников;- планирование для ремонтного персонала;- подтверждение факта выполненной работы сотрудников и проставление допуска;- составление отчетности;- подтверждение спланированного дня сотрудника;- фильтрация выводимых данных.
Ремонтный персонал	<ul style="list-style-type: none">- планирование задач;- изменение статуса спланированной задачи;- фильтрация выводимых данных.
Оперативный персонал	<ul style="list-style-type: none">- добавление допуска для ремонтного персонала;- фильтрация выводимых данных.
Администратор	<ul style="list-style-type: none">- изменение данных;- подтверждение/отмены регистрации пользователей;- изменение данных пользователей, кроме пароля и почты;- удаление пользователей.

2.2.1 Работа руководителя с информационной системой

Для руководителя будет доступна большая часть функционала представленной ИС.

При планировании задач руководителю будет необходимо из выпадающего списка выбрать сотрудника, для которого планируется рабочий день. Выбрать на какую дату планируется день. А также прописать задачи и описание задачи на выбранное рабочее время. Задачи могут задаваться тремя способами: ввод вручную, выбор из невыполненных ранее задач или из внешнего файла. И имеется возможность автоматического планирования рабочего дня для сотрудника.

В выведенной таблице с информацией спланированных задач руководитель может изменять данные в полях (рис. 5) «допуск» и «примечание». При подтверждении о выполненной работе ремонтного персонала, руководитель может выставить значение, что задача была: подтверждена (задача больше не будет выводиться при планировании), не подтверждена (задача будет выводиться при планировании и будет закреплена за данным сотрудником), перенесена (аналогично тому, как и не подтверждена), снята (задача будет выводиться при планировании, но не будет закреплена для конкретного сотрудника, а будет видна для любого ремонтного персонала) и закрыто (задача больше не показывается при планировании из-за ее не актуальности). А также предоставлять допуск, после прохождения инструктажа, к работе, если это необходимо.

Подтверждение спланированного дня сотрудника необходим для того, чтобы руководитель мог увидеть, что запланировал себе исполнитель и при необходимости внести незначительные изменения и подтвердить, или же подтвердить план без внесения изменений.

Следующий функционал ИС – фильтрация данных, является хорошим помощником для любого пользователя. Благодаря фильтрации руководитель может из массы данных вывести, только ту, которая необходима ему в данный момент. К примеру, посмотреть, как выполнял задачи сотрудник за

определенный период работы, вывести информацию только тех задач, которые не были выполнены и т.д. Что значительно облегчает и ускоряет поиск необходимой информации.

При выборе руководителем функции составления отчетности (экспорт в Excel), будет сформирована таблица в виде Excel-файла. Далее начнется автоматическое скачивание данного файла и сохранение его на ПЭВМ. Таблица в скаченном файле будет сформирована абсолютно такая же, как и на главной странице WEB-приложения.

2.2.2 Работа ремонтного персонала с информационной системой

Планирование происходит аналогично, как и руководителем, за исключением того, что спланированный день отправляется на подтверждение руководителю.

После подтверждения спланированного дня руководителем, данные заносятся в общую таблицу, где исполнитель может менять данные таких полей (рис. 5), как: результат (заносятся исполнителем отчет о проделанной работе, задача была выполнена или нет), обоснование (заполняется по какой причине данная задача не была выполнена) и часы учета (выставляется количество часов, потраченных на выполнение задачи).

Так же доступен функционал фильтрации массива данных для вывода необходимой информации в таблицу.

2.2.3 Работа оперативного персонала с системой

Для данной группы пользователей нет необходимости в планировании задач, оперативный персонал лишь заполняет наименование допуска (если оно отличается от допуска выставленным руководителем) к выполнению работ и выставляет часы, затраченные на выполнение работ с допуском.

Для более удобного и быстрого поиска исполнителя, для которого нужно внести изменения в таблицу, так же имеется возможность использования фильтрации данных. В сформированной таблице оперативный персонал может изменять только данные в таких полях, как (рис. 5) «факт. часы с допуском» и «наименование допуска».

2.2.4 Работа администратора с информационной системой

Группа администратор необходимо для того, чтобы подтверждать или отменять регистрацию нового пользователя в системе, редактировать данные пользователей (при необходимости), а именно:

- вносить изменения в фамилию, имя и отчество (необходимо если вдруг по какой-то причине сотрудник неверно ввел данные при регистрации);
- изменять роль пользователя (к примеру, исполнителя назначили руководителем или наоборот и для этого нужно будет внести изменения);
- менять руководителя у ремонтного персонала;
- удалить пользователя.

В таблице спланированных данных администратор имеет доступ к изменению всех полей. Так же доступен функционал фильтрации данных.

Следующим шагом рассмотрим предварительное проектирование подсистем в отдельности, для реализации общей совокупности информационной системы.

2.3 Подсистема добавления нового пользователя

Добавление пользователя (сотрудника) – данный функционал позволяет регистрировать нового пользователя, после чего позволяет проходить авторизацию по введенным данным и попадать в систему планирования и учета ремонтных работ. Благодаря данному элементу доступ к данным и вход в систему посторонним лицам невозможен.

Для завершения регистрации нового сотрудника необходимо заполнить шесть полей:

- фамилия;
- имя;
- отчество;
- почта (является логином пользователя);
- выбор прав доступа (руководитель, исполнитель, ремонтный персонал);

- выбор руководителя (данное поле появляется если в поле права доступа было выбрано «исполнитель», необходимо выбрать ФИО руководителя из выпадающего списка);

- пароль (секретное, кодовое слово необходимое при авторизации);

- подтверждения пароля (исключает ошибочное написание кодового слова при регистрации).

Стоит заметить, что при регистрации нового пользователя в целях безопасности пароль сохраняется в БД в зашифрованном виде.

2.4 Подсистема авторизации пользователя

Данная подсистема предназначена для проверки подлинности пользователей и разграничению прав доступа. Процедура идентификации осуществляется по двум полям: ввода почты и пароля. При правильном заполнении соответствующих полей запускается процедура авторизации, в противном случае выводится об ошибке, связанной с отказом в доступе.

Стоит заметить, чтобы пройти идентификацию пользователю успешно, необходимо дождаться подтверждения регистрации администратором в системе.

Процедура авторизации выполняет функцию наделения пользователя правами доступа к защищенной информации.

2.5 Подсистема планирования

Подсистема является ядром системы и позволяет непосредственно планировать задачи на рабочий день. Данный функционал возложен на следующие группы пользователей: руководитель и ремонтный персонал.

В случае попытки воспользоваться подсистемой пользователям, не обладающих данными правами, будет выведено сообщение об ограничении действия.

Как говорилось выше в данной подсистеме происходит заполнение предложенной формы, в которой необходимо выбрать сотрудника (для которого планируется рабочий день), дату планирования и сами задачи с указанием времени (которое отводится на задачу).

В подсистеме для пользователей, планирующих задачи, существует четыре вида добавления задач и их описание:

- автоматическое планирование рабочего дня;
- ввод происходит вручную;
- выбор из предложенного списка, задачи, которые не были выполнены ранее;
- выбор из предложенного списка, задачи, которые были сформированы из отдельного файла (к примеру, журнал дефектов).

При выборе автоматического планирования формируется список задач для выбранного сотрудника (или им самим) по следующим этапам: сначала подтягиваются задачи с примечаниями «не подтверждено» и «перенесено», после добавляются задачи с примечанием «снято», которые находятся в общем доступе и уже последним шагом задачи из файла.

2.6 Подсистема составления отчетности

Использовать функционал данной подсистемы могут только пользователи с правами доступа руководителя. На странице формируется таблица со всеми данными (так же имеется возможность эти данные отфильтровать) и при нажатии на кнопку «Экспорт вExcel» формируется таблица в excel файле.

2.7 Подсистема фильтрации данных

Данная подсистема доступна для всех авторизованных пользователей. Эта подсистема является хорошим функционалом для систем с большим объемом данных. Благодаря этой функции можно из всего массива данных вывести только ту, которая необходима. Выбор фильтров происходит посредством заполнения предложенной формы.

Настройка фильтра возможна как выбором одного из полей, так и нескольких сразу. Возможность отфильтровать по таким полям, как (рис. 5): по дате, по сотруднику, по времени, по задаче, по описанию задачи, по результату выполнения, по примечанию и по учету ремонта.

РАЗРАБОТКА СТРУКТУРЫ БАЗЫ ДАННЫХ

Для реализации ИС, разработка базы данных является первостепенной задачей в построении решения. Необходимо максимально точно выделить подзадачи системы, чтобы составить правильно необходимые таблицы и связи между ними.

Представленное решение в данной работе будет реализована с помощью трех основных таблиц («Пользователи», «Дата планирования» и «Задачи») и две второстепенных («Группа пользователей» и «Примечания»).

Сущность «Пользователи» содержит информацию об сотрудниках зарегистрированных в ИС.

Сущность «Дата планирования» содержит информацию о спланированном рабочем дне, на какую дату и для кого.

Сущность «Задачи» содержит информацию о спланированных задачах по отдельности.

Сущность «Группа пользователей» содержит информацию о правах доступа пользователей в ИС.

3.1 Атрибуты сущности «Пользователи»

В данной таблице будут храниться строки, содержащие необходимую информацию о пользователе (таблица 3).

Таблица 3 – Атрибуты сущности «Пользователи»

Название атрибута	Описание атрибута	Диапазон значений	Пример
1	2	3	4
<u>Код пользователя (id)</u>	Уникальный идентификатор пользователя	>0	1;2;3...n
Имя (first_name)	Имя пользователя	-	Василий
Фамилия (last_name)	Фамилия пользователя	-	Иванов

Продолжение таблицы 3

1	2	3	4
Отчество (middle_name)	Отчество пользователя	-	Сергеевич
Код группы (id_group)	Уникальный идентификатор группы пользователей	>0	1;2;3...n
Почта (email)	Почта пользователя (логин для авторизации в ИС)	-	<u>example@mail.ru</u>
Пароль (password)	Пароль пользователя для авторизации в ИС	-	wQeSWv23dw
Код руководителя (id_leader)	Уникальный идентификатор пользователя	>0	1;2;3...n
Администратор (is_admin)	Указывает на то является ли пользователь администратором	ИСТИНА или ЛОЖЬ	0;1
Подтверждение (confirmation)	Указывается на то была ли подтверждена регистрация пользователя	ИСТИНА или ЛОЖЬ	0;1

Заполнение данной таблицы происходит при регистрации пользователя на сайте (п.4.2). Имеются следующие поля:

- id (идентификатор пользователя, является счетчиком и заполняется автоматически при добавлении записи в базу);

- first_name, last_name и middle_name (формат данных полей – текст, хранится имя, фамилия и отчество пользователя);

-id_group (формат поля – число, данное поле связано с полем «id»из таблицы «Группа пользователей», таблица 4);

- email (тип данных данного поля – текст, содержится почта пользователя);

- password (тип данных – текст, хранит пароль пользователя, также данное значение поля шифруется и заносится в таблицу).

Таблица 4 – Атрибуты сущности «Группа пользователей»

Название атрибута	Описание атрибута	Диапазон значений	Пример
<u>Код группы (id)</u>	Уникальный идентификатор группы	>0	1;2;3...n
Название (title_group)	Наименование прав доступа	-	«Руководитель», «Исполнитель», «Оперативный персонал»

3.2 Атрибуты сущности «Дата планирования»

Данная сущность хранит в себе информацию о дате планирования и исполнителе (таблица 5).

Таблица 5 – Атрибуты сущности «Дата планирования»

Название атрибута	Описание атрибута	Диапазон значений	Пример
1	2	3	4
<u>Код даты планирования (id)</u>	Уникальный идентификатор даты планирования	>0	1;2;3...n

1	2	3	4
Дата (date)	Значение даты планирования	Дата/время	20.06.2018
Код пользователя (id_user)	Уникальный идентификатор пользователя	>0	1;2;3...n
Подтверждение (accepted)	Указывается на то был ли подтвержден план	ИСТИНА или ЛОЖЬ	0;1

Содержит следующие поля для заполнения:

- id (идентификатор даты планирования, является счетчиком и заполняется автоматически при добавлении записи в базу);
- date (тип данных поля является дата, хранит в себе дату планирования на выполнение задач);
- id_user(тип данных – число, хранится значение уникального идентификатора пользователя из таблицы «Пользователи»).

3.3 Атрибуты сущности «Задачи»

Сущность «Задачи» является ключевой, она несет в себе основную часть информации для работы с ИС (таблица 6).

Таблица 6 – Атрибуты сущности «Задачи»

Название атрибута	Описание атрибута	Диапазон значений	Пример
1	2	3	4
<u>Код задачи (id)</u>	Уникальный идентификатор задачи	>0	1;2;3...n

1	2	3	4
Код даты планирования (id_dateplan)	Уникальный идентификатор даты планирования	>0	1;2;3...n
Время начала (time_with)	Фактическое время начала выполнения задачи	Время	08:00; 09:00...16:00
Время окончания (time_before)	Фактическое время окончания выполнения задачи	Время	09:00; 10:00...17:00
Задача (title)	Сама задача	-	Настройка и конфигурирование технологической сети
Описание (description)	Описание задачи	-	маршрутизация и доступ к контроллерам АСР
Результат (result)	Результат о проделанной задаче	«Выполнено», «Не выполнено»	выполнено; не выполнено
Обоснование (rationale)	Обоснование о не выполненной задаче	-	в процессе

Продолжение таблицы 6

1	2	3	4
Код примечания (id_notation)	Уникальный идентификатор примечания	>0	1;2;3...n
Допуск (admission)	Наименование допуска	-	ИВС
Учет ремонта	Учет ремонта	«да» или «нет»	да; нет
Часы учета (accounting_hours)	Количество часов, затраченных на задачу	>=0; <=8	0;1;2...8
Плановое время (plan_time)	Предварительное кол- во часов на выполнение задачи	>=1, <=8	1;2;3...8
Тип времени (type_time)	Тип времени спланированной задачи (1 – обычное время, 2- внеплановое, 3 – офиц. рем.)	1, 2, 3	1;2;3
Фактические часы с допуском (time_with_admiss)	Количество часов выполнения задачи с допуском	>=1, <=8	1;2;3...8
Наименование допуска (name_admiss)	Наименование допуска, выставленное оперативным персоналом	-	ИВС

Приведены следующие столбцы:

- id (данное поле является стандартным и обязательным столбцом для всех таблиц, описано выше);
- id_dateplan (тип данных – число, данное поле имеет связь с полем «id» из таблицы «date_plan», значение записывается автоматически при добавлении плана на странице п.4.3);
- time_with (тип – время, задается время начала выполнения задачи);
- time_before (тип – время, задается время окончания выполнения задачи);
- title (формат поля является текст, ограничений на ввод значений нет, хранятся непосредственно сами задачи);
- description (формат поля – текст, ограничений не имеется, хранится описание задачи);
- result (формат – текст, хранит результат проделанной работы, возможен ввод текста только «выполнено» и «не выполнено»);
- rationale (текстовый формат столбца, ограничений на ввод нет, хранится объяснение исполнителя о не выполненной задаче);
- id_notation (поле является целым беззнаковым, имеет связь с атрибутом «Код примечания» из сущности «Примечания», таблица 7);
- admission (формат – текст, хранится ФИО исполнителя на допуск);
- repair_accounting (тип данных данного поля – boolean, возможен ввод значений только 0 – «нет» и 1 – «да», хранит информацию является ли задача ремонтной);
- accounting_hours (численный тип данных, содержится информацию затраченных часов на выполнение задачи исполнителем);
- plan_time(тип данных данного поля является целое число, хранится значение планируемого времени на выполнение задачи).

Таблица 7 – Атрибуты сущности «Примечания»

Название атрибута	Описание атрибута	Диапазон значений	Пример
<u>Код примечания (id)</u>	Уникальный идентификатор примечания	>0	1;2;3...n
Заголовок	Наименование примечания	«Подтверждено», «Не подтверждено», «Перенесено», «Снято» и «Закрыто»	Снято

4 РАЗРАБОТКА СТРУКТУРЫ ИНТЕРФЕЙСА ПОЛЬЗОВАТЕЛЯ

Структура данного WEB– приложения является двухуровневым (рис. 10).

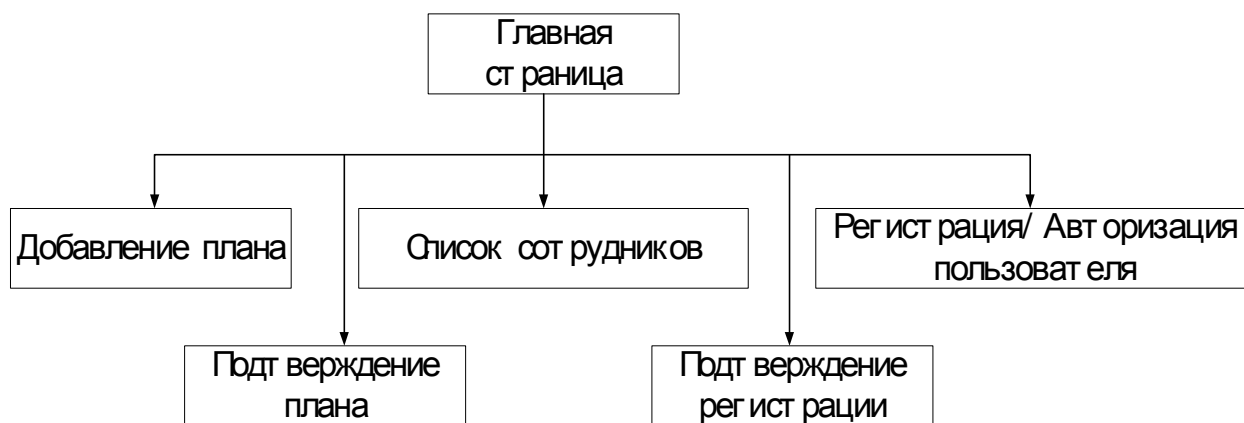


Рисунок 10 – Структура сайта предложенного решения

4.1 Главная страница

Основное окно ИС является главная страница, на которой будет выводиться вся необходимая информация для авторизованных пользователей. Если страницу посетит неавторизованный пользователь, то будет выведено сообщение о невозможности просмотра информации (рис. 11). Для авторизованного пользователя (руководителя или исполнителя) будет выведено на монитор:

- таблица данных;
- кнопка добавления нового плана (для руководителя и исполнителя);
- кнопка перехода на просмотр списка сотрудников (для руководителя);
- кнопка перехода на список спланированных задач (для руководителя);
- кнопка перехода на список пользователей ИС (для администратора);
- кнопка настройки фильтра (для всех авторизованных пользователей);
- кнопка регистрации.

Вид главной страницы представлен в приложении Б.

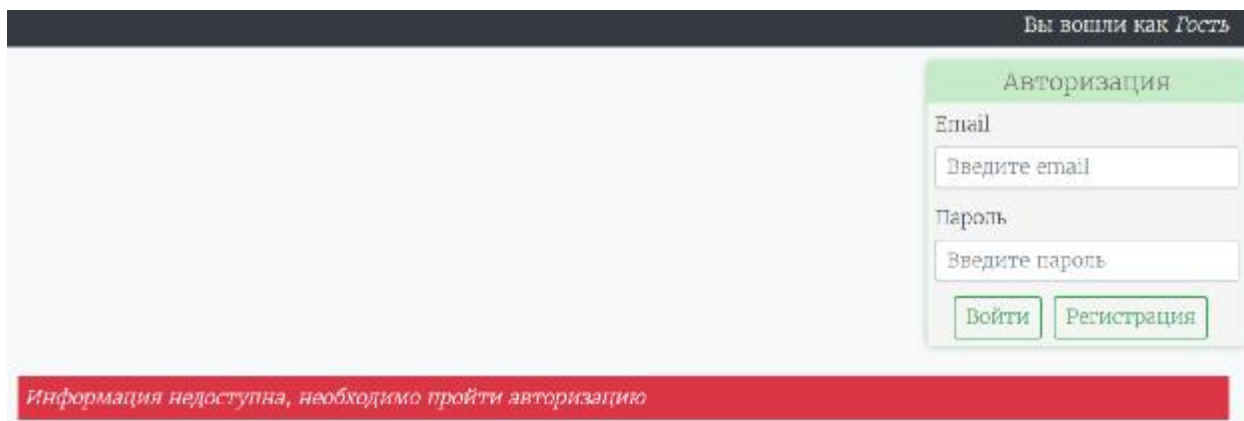


Рисунок 11 – Вывод сообщения о доступе к данным

При нажатии на кнопку «Регистрация» открывается форма (п.4.2) на заполнение для регистрации в информационной системе.

При нажатии на кнопку «Добавить план» открывается новая страница «Добавление плана» (п.4.3).

При нажатии на кнопку «Список сотрудников» открывается новая страница «Список сотрудников» (п.4.4).

При нажатии на кнопку «Список планирований» открывается новая страница, где выводится список спланированных задач ремонтным персоналом, которые не были подтверждены руководителем (п.4.5).

После нажатия администратором на кнопку «Список пользователей» открывается страница, на которой выводится таблица всех зарегистрированных пользователей в ИС (п.4.6).

После нажатия на кнопку «Настроить фильтр» раскрывается форма (рис. 12), для выбора на вывод в таблицу нужной информации. Для вывода информации под настроенный фильтр, пользователю необходимо заполнить как минимум одно поле из следующих:

- выбор даты (данное поле является выпадающим списком, которое формируется автоматически, выводит в список даты добавленного плана);
- выбор сотрудника (выпадающий список, заполняющийся автоматически всеми исполнителями из БД);

- время фактическое (нужно указать значение времени «с» или «до», а также выбор иного из выпадающего списка: «внеплан.» и «офиц. рем.»);
- задача (значение данного поля выбирается из выпадающего списка, в котором перечислены все возможные задачи, которые когда-либо были занесены в базу);
- описание (выпадающий список, предлагается выбор из всех описаний задач, внесенных в базу);
- результат выполнения (выпадающий список, возможный выбор только из «выполнено» и «не выполнено»);
- примечание (выпадающий список, возможные значения полей «подтверждено», «не подтверждено», «перенесено» и «снято»);
- учет ремонта (выпадающий список со значениями «да» и «нет»).

Рисунок 12 – Настройка фильтра на вывод данных в таблицу

Так же имеется кнопка «Экспорт в Excel» (рис. 13), которая позволяет сохранить в виде таблицы Excel выведенную таблицу на монитор.

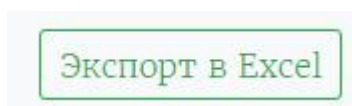


Рисунок 13 – Кнопка «Экспорт в Excel»

Дата	Время	Приниматель	Тариф	Описание	Баланс	Объем нажима	Промежуток	Длительность	Участники	Средняя скорость	Фактическая скорость	Время
				Инициализация мобильного устройства, установка приложения, регистрация в системе	1000 руб	1000 руб	1000 руб	1000 руб	1000 руб	1000 руб	1000 руб	1000 руб
				Инициализация мобильного устройства, установка приложения, регистрация в системе	1000 руб	1000 руб	1000 руб	1000 руб	1000 руб	1000 руб	1000 руб	1000 руб
				Инициализация мобильного устройства, установка приложения, регистрация в системе	1000 руб	1000 руб	1000 руб	1000 руб	1000 руб	1000 руб	1000 руб	1000 руб
				Инициализация мобильного устройства, установка приложения, регистрация в системе	1000 руб	1000 руб	1000 руб	1000 руб	1000 руб	1000 руб	1000 руб	1000 руб
				Инициализация мобильного устройства, установка приложения, регистрация в системе	1000 руб	1000 руб	1000 руб	1000 руб	1000 руб	1000 руб	1000 руб	1000 руб
				Инициализация мобильного устройства, установка приложения, регистрация в системе	1000 руб	1000 руб	1000 руб	1000 руб	1000 руб	1000 руб	1000 руб	1000 руб
				Инициализация мобильного устройства, установка приложения, регистрация в системе	1000 руб	1000 руб	1000 руб	1000 руб	1000 руб	1000 руб	1000 руб	1000 руб
				Инициализация мобильного устройства, установка приложения, регистрация в системе	1000 руб	1000 руб	1000 руб	1000 руб	1000 руб	1000 руб	1000 руб	1000 руб
				Инициализация мобильного устройства, установка приложения, регистрация в системе	1000 руб	1000 руб	1000 руб	1000 руб	1000 руб	1000 руб	1000 руб	1000 руб

Рисунок 14 – Полученная таблица в Excel

4.2 Форма регистрации пользователя

При нажатии на кнопку «Регистрация» открывается форма (рис. 15), которую необходимо заполнить для успешной регистрации нового пользователя в ИС. После чего необходимо ждать подтверждение регистрации для дальнейшего пользования функционала ИС.

Рисунок 15 – Форма добавления сотрудника

Чтобы регистрация прошла успешно необходимо заполнить следующие поля:

- фамилия, имя и отчество (возможен ввод только буквенных значений, максимальная длина строки 24 символа);
- email (возможен ввод, как латинских символов, так и цифр, обязательная проверка на символ «@», также является логином для авторизации в системе);
- пароль (может состоять из любых символов, кроме кириллицы и пробелов);

- выбор прав доступа (возможен выбор: руководитель, исполнитель и оперативный персонал);
- выбор руководителя (данное поле появляется при выборе прав доступа как исполнитель);
- повторение пароля(введенная строка должна совпадать со значением введенное в поле «пароль»).

4.3 Страница добавления плана

На данной странице имеется форма для добавления задач для исполнителя (рис. 16). Данная форма может заполняться непосредственно, как руководителем, так и исполнителем.

Полный вид данной страницы представлен в приложении В.

The screenshot shows a web interface for adding tasks. The main form has the following elements:

- Button: Спланировать автоматически
- Field: Выбор сотрудника: (dropdown)
- Field: Дата: 24.05.2018
- Section: Время факт.. containing two time input fields (08:00, 09:00), a task description dropdown (Тех.гра, анализ проектов, оценки эффективности производственной работы), a 'Стоимость:' field, and a '1 час' dropdown.
- Button: время
- Button: Добавить план

On the right side, there are two sections:

- Неисполненные задачи (общие): A table with columns for task ID, status, and description.
- Журнал дефектов (из документа): A table with columns for defect ID, status, and description.

Рисунок 16 – Форма добавления задач

Для того, чтобы добавить план необходимо заполнить следующие поля, обязательные для заполнения:

- выбор сотрудника (значение данного поля выбирается из выпадающего списка, который формируется автоматически из пользователей с базы данных);
- дата (автоматически заполняется текущей датой, но также есть возможность изменения даты вручную);
- время фактическое (данное поле имеет несколько строк и пять столбцов. В первом столбце указывается время начала выполнения задачи, а

во втором столбце время окончания выполнения задачи. Третий столбце непосредственно пишется сама задача на выполнение, четвертый столбец – описание задачи, обычно пишутся подзадачи на выполнение. В пятом столбце выбирается из выпадающего списка количество планируемых часов на выполнение задачи).

При загрузке страницы в поле «время фактическое» формируется автоматически одна строка, если необходимо добавить еще строку, то нужно нажать на кнопку «время +» и будет добавлена одна дополнительная строка, а для удаления лишней строки имеется справа от четвертого столбца кнопка «X».

После заполнения всех полей и нажатие на кнопку «Добавить план», происходит проверка на правильность введенных значений и, если нет каких-либо замечаний, происходит добавление задач в базу с меткой не подтверждена руководителем. После у руководителя появится данное предложение спланированного дня, которое необходимо либо подтвердить, либо отложить на потом.

4.4 Страница просмотра списка сотрудников

На данной странице выводится список всех сотрудников в подчинении у руководителя в виде таблицы (рис. 17).



The screenshot shows a web application interface. At the top right, there is a user profile section with the text "Добро пожаловать, *Нещеменко К.Д.*" and a "Выйти" button. Below this is a breadcrumb trail: "Главная страница / Список сотрудников". The main content is a table with the following data:

№	фамилия	Имя	Отчество	Почта	Последнее планирование
1	Ивапов	Николай	Николаевич	inn@mail.ru	16.06.2018

Рисунок 17 – Таблица вывода списка сотрудников

4.5 Страница просмотра спланированных задач

На странице выводится список спланированного дня ремонтного персонала (рис. 18), которые еще не были подтверждены руководителем на исполнение. Полный вид страницы представлен в приложении Г.

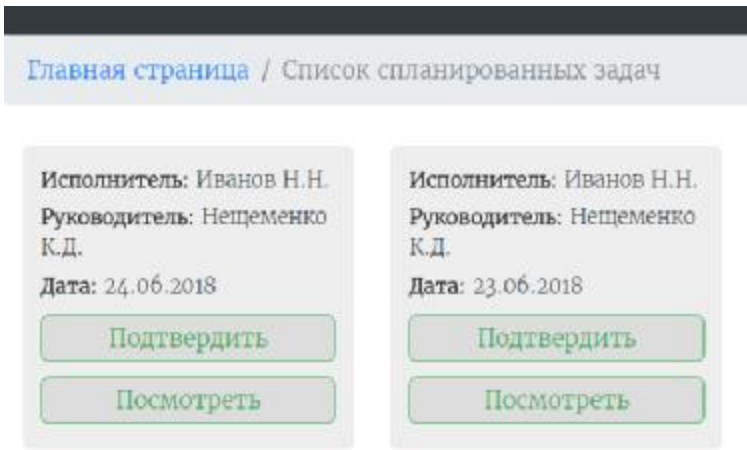


Рисунок 18 – Список спланированных дней

В данных блоках содержится информация о исполнителе (кем был день спланирован), о его руководителе (только он может подтвердить план на исполнение) и о дате планирования (на какое число был спланирован день). Также имеется возможность руководителю посмотреть (рис. 19), что было спланировано сотрудником и по необходимости изменить задачи и подзадачи (при необходимости написать сразу допуск к задаче).

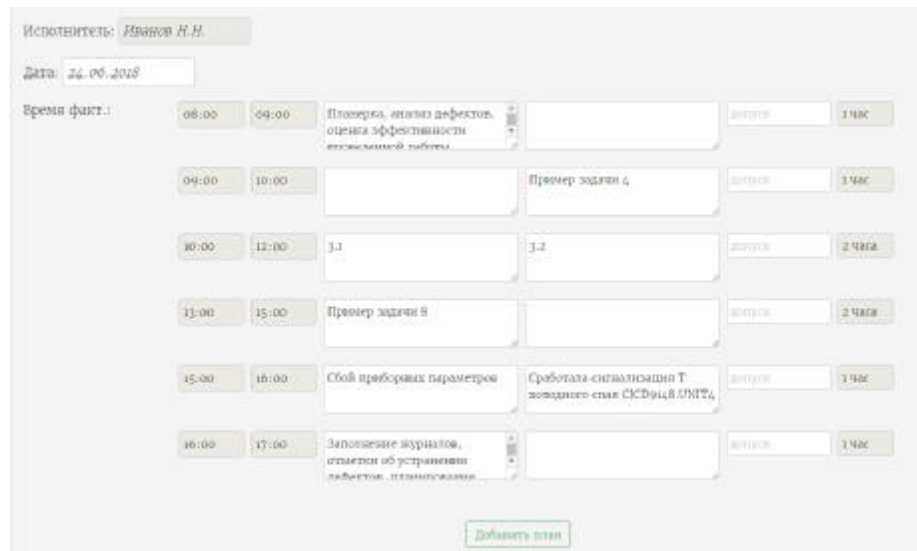


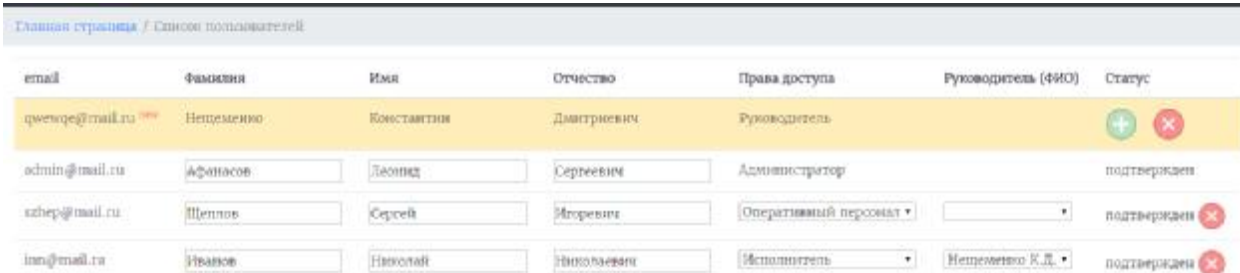
Рисунок 19 – Просмотр спланированного дня ремонтного персонала

После нажатии на кнопку «Добавить план» спланированный день получает статус подтверждено и вносятся все изменения, которые были

внесены руководителем, далее данные будут выводиться в общую таблицу, где исполнитель уже увидит свой спланированный день.

4.6 Страница зарегистрированных пользователей

Страница для просмотра доступна только администраторам. На данной странице выводится весь список зарегистрированных пользователей в ИС (рис. 20). Полный вид странице представлен в приложении Д.







email	Фамилия	Имя	Отчество	Права доступа	Руководитель (4960)	Статус
qwewqe@mail.ru	Ищепемяю	Константин	Дмитриевич	Руководитель		 
admin@mail.ru	<input type="text" value="Афанасов"/>	<input type="text" value="Леонид"/>	<input type="text" value="Сергеевич"/>	Администратор		подтвержден
szher@mail.ru	<input type="text" value="Щеплов"/>	<input type="text" value="Сергей"/>	<input type="text" value="Игоревич"/>	Оперативный персонал		подтвержден 
inn@mail.ru	<input type="text" value="Иванов"/>	<input type="text" value="Николай"/>	<input type="text" value="Николаевич"/>	Исполнитель	Ищепемяю К.Д.	подтвержден 

Рисунок 20 – Список зарегистрированных пользователей

Сначала списка выводятся пользователи, регистрация которых не была подтверждена, затем все остальные. Администратор может не подтвержденных пользователей удалить либо подтвердить регистрацию. Как видно из рис. 20 администратор может изменять данные пользователей, менять права доступа и руководителя, а также удалять пользователя.

5 ПРОГРАММНАЯ РЕАЛИЗАЦИЯ

Этап реализации информационной системы разбит на три части:

- выбор языка программирования для написания ИС;
- создание программных модулей (подсистем ИС);
- написание запросов к сущностям СУБД.

На рис. 21 отображены основные компоненты ИС и их взаимодействие.

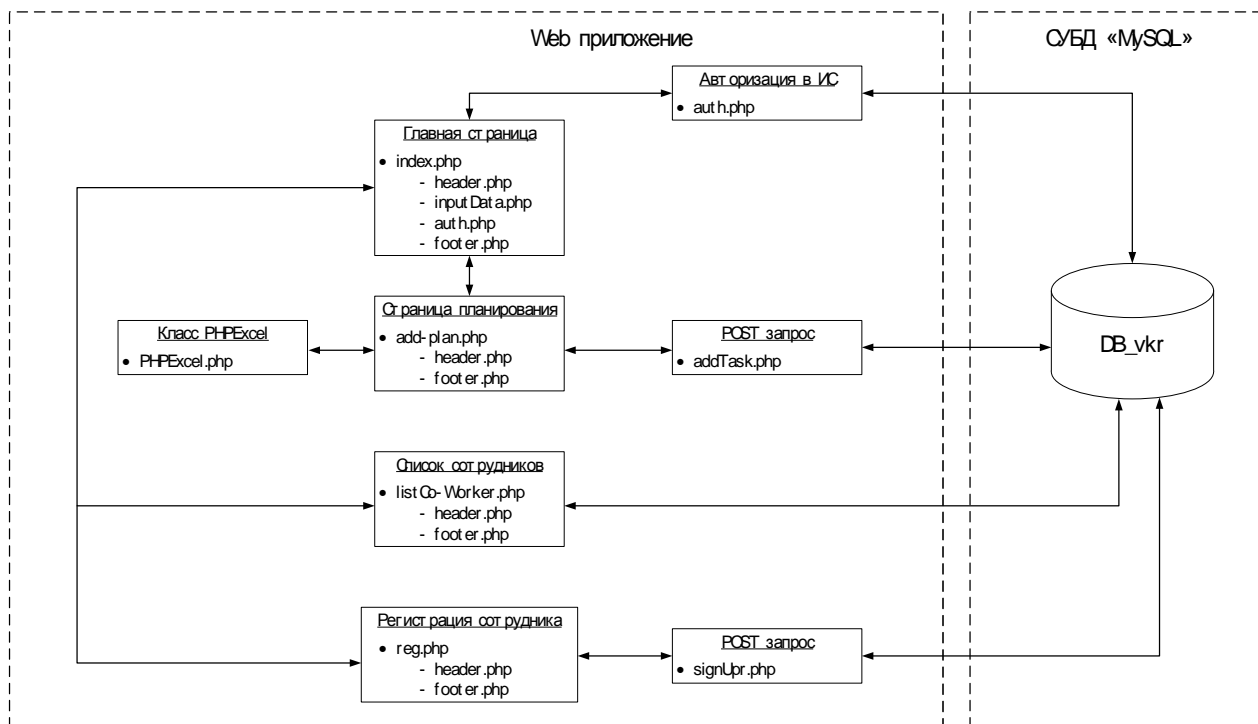


Рисунок 21 – Логическая структура программы

Для хранения данных была создана БД «DB_vkr» в СУБД «MySQL», принимающая запросы от приложения и предоставляющая необходимые данные.

Все приложение разделено на модули, каждый из которых представляет собой окно. Каждое окно состоит из основного файла и файлов модулей. Класс PHP Excel необходим для работы с excel файлами. Данный класс используется для считывания задач на выполнения с файла, а также для составления отчетности.

5.1 Выбор языка программирования

Информационная система состоит из двух частей: серверной и клиентской. Имеется возможность использовать один и тот же язык программирования, однако это является самым неоптимальным вариантом.

Для серверной части будет использоваться PHP [8]. PHP – скриптовый язык программирования общего назначения, интенсивно применяющийся для разработки веб-приложений. В настоящее время поддерживается подавляющим большинством хостинг-провайдеров и является одним из лидеров среди языков программирования, применяющихся для создания динамических веб-сайтов.

Для клиентской части будут использоваться следующие языки:

- HTML [9] (отангл. HyperTextMarkupLanguage – «язык разметки гипертекста») – стандартный язык разметки документов во всемирной паутине;

- JavaScript[10] – объектно-ориентированный скриптовый язык программирования. JS поддерживается всеми существующими браузерами и является стандартом для современных интерактивных веб-приложений.

Так же необходимо определиться с выбором Web-сервера. Web-сервер – это сервер, принимающий HTTP-запросы от клиентов, обычно веб-браузеров, и выдающим им HTTP-ответы, обычно вместе с HTML-страницей, изображение, файлом, медиа-поток или другими данными. Web-серверы – основа всемирной паутины.

На данный момент существует всего два Web-сервера, которые позволяют реализовать совместную работу ранее выбранных языков программирования:

- Nginx [11] – высокопроизводительный HTTP-сервер, предназначенный в основном для раздачи клиентам статического контента (изображений, js-файлов, css-стилей и т.д.). В основе лежит технология неблокирующих соединений, что при большом количестве одновременных подключений существенно экономит ресурсы сервера.

- ApacheHTTP-сервер[12] – свободный веб-сервер. Apache является кроссплатформенным ПО, поддерживает операционные системы Linux, BSD, Mac OS, Microsoft Windows, Novell NetWare, BeOS. Основными достоинствами Apache считаются надёжность и гибкость конфигурации. Он позволяет подключать внешние модули для предоставления данных, использовать СУБД для аутентификации пользователей, модифицировать сообщения об ошибках и т. д.

Возможна совместная работа двух Web-серверов: nginx обрабатывает запросы на раздачу статичного контента, в то время как apache курирует выполнение серверных скриптов.

5.2 Реализация подсистем информационной системы

Для функционирования ИС были определены и реализованы алгоритмы подсистем, решающие отдельные подзадачи данной системы, поставленные перед ней.

5.2.1 Реализация подсистемы авторизации

Функциональное назначение: защита доступа ко всем данным от посторонних лиц, а также использование функционала ИС.

Алгоритм авторизации пользователя представлен на рис. 22.

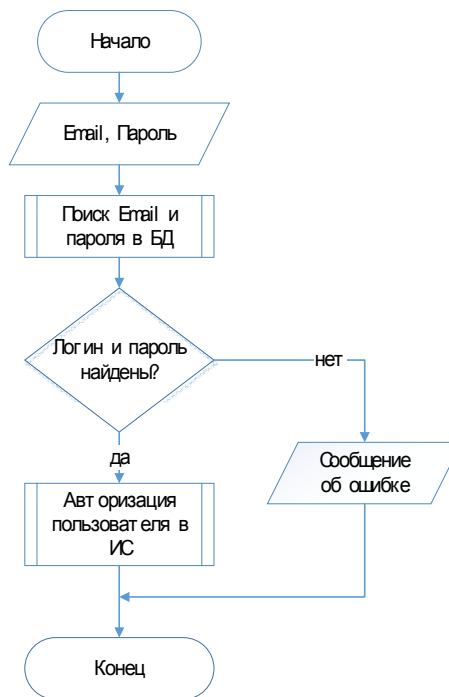


Рисунок 22 – Алгоритм авторизации в ИС

Пользователь вводит данные в поля почта (Email) и пароль, которые затем сравниваются с данными email и пароля сотрудника, хранящимися в БД. Если введенные данные совпадают с данными из БД, то происходит авторизация пользователя и предоставляется доступ к функционалу ИС, согласно правам доступа. В противном случае выводится сообщение об ошибке. Возможные сообщения ошибки: «пользователь с данным e-mail не найден», «пароль введен неверно».

Часть реализации программного кода представлена на рис. 23.

```

1  <?php
2      $data = $_POST;
3      if (isset($data['logIn'])) {
4          $errors = "";
5          if ($data['email'] == "" || $data['password'] == "")
6              $errors = "Заполните все поля!";
7          else {
8              $user = getUser($data['email'],null);
9              if ($user) {
10                 if (password_verify($data['password'], $user['password'])) {
11                     $_SESSION['logged_user'] = $user; ?>
12                     <script language="JavaScript">
13                         window.location.href = "/"
14                     </script>
15                     <?php
16                     }
17                 else {
18                     $errors = "Пароль введен неверно.";
19                 }
20             }
21             else {
22                 $errors = "Пользователь с данным email не найден.";
23             }
24         }
25     }
26     ?>

```

Рисунок 23– Часть программного кода авторизации

5.2.2 Реализация подсистемы регистрации

Данный функционал позволяет регистрироваться в ИС, тем самым позволяет разделить пользователей на разные группы доступа и использовать авторизацию в ИС для защиты данных от третьих лиц.

Алгоритм добавления сотрудника в БД представлен на рис. 24.

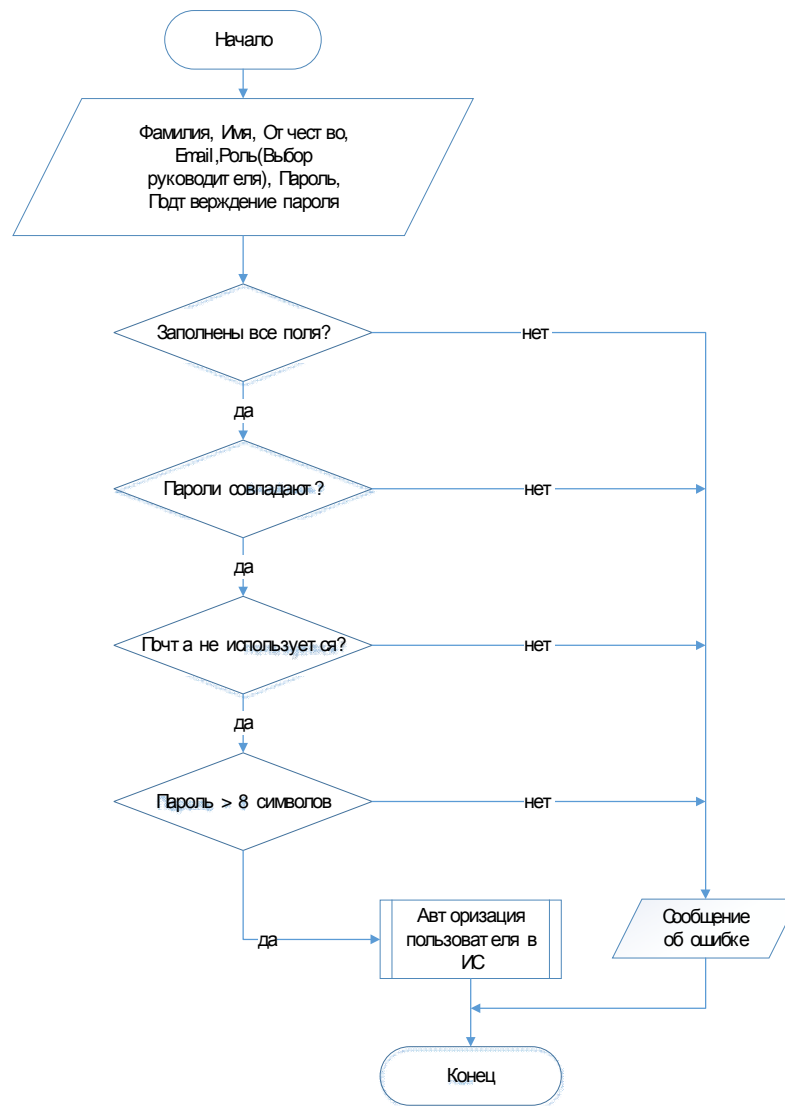


Рисунок 24 – Алгоритм добавления сотрудника

После нажатия пользователем на кнопку «Зарегистрироваться» происходит проверка на заполнение значением всех полей. Если проверка проходит успешно и все поля заполнены, то осуществляется дальнейшая проверка на правильность ввода пароля (совпадают ли значение введенные в поле «пароль» с полем «повторите пароль»). При несоответствиях значений появится сообщение об ошибке – «пароли не совпадают». После успешной проверки происходит дальнейшая обработка значения введенное в поле «почта» на то, чтобы пользователь с данной почтой не был уже зарегистрирован. Если было найдено совпадение почты в БД, то выведется соответствующее сообщение об ошибке – «пользователь с данной почтой

уже существует». Далее проверяется чтобы введенный пароль содержал больше 8 символов, фамилия, имя и отчество содержали только символьные значения.

По прохождению успешно всех проверок новый пользователь заносится с БД с теми данными, которые он ввел в форму при регистрации. Для успешной авторизации в ИС потребуются данные почты и пароля, и ожидать подтверждения регистрации администратором.

Часть программного кода представлена на рис. 25.

```
if ($password != $data['r_pssw']) {
    $mssg = 'Пароли не совпадают';
    return;
}
else if (getUser($email,null)) {
    $mssg = 'Данная почта уже используется';
    return;
}
else {
    $new_pssw = password_hash($password, PASSWORD_DEFAULT);
    if (addUser($first_name, $middle_name, $last_name, $email, $new_pssw, $id_leader))
    {
        $mssg = 'Сотрудник был успешно добавлен. Перейти на <a href="/">главную
        страницу</a> или на страницу <a href="/listCo-worker.php"> списка
        сотрудников</a>';
        return;
    }
    else {
        $mssg = 'Что-то пошло не так. Проверьте подключение к интернету';
        return;
    }
}
}
```

Рисунок 25– Часть программного кода добавления сотрудника

5.2.3 Реализация подсистемы планирования ремонтных работ

Данный функционал информационной системы позволяет спланировать задачи для ремонтного персонала на рабочий день. Для пользователей с правами доступа ремонтного персонала после успешного спланированного дня, план попадает на рассмотрение руководителю, который в свою очередь принимает его или вносит изменения и добавляет. Также после успешного добавления плана нет возможности дальнейшего добавления задач на выполнение и внесения изменений.

Алгоритм планирования рабочего дня для ремонтного персонала представлен на рис. 26.

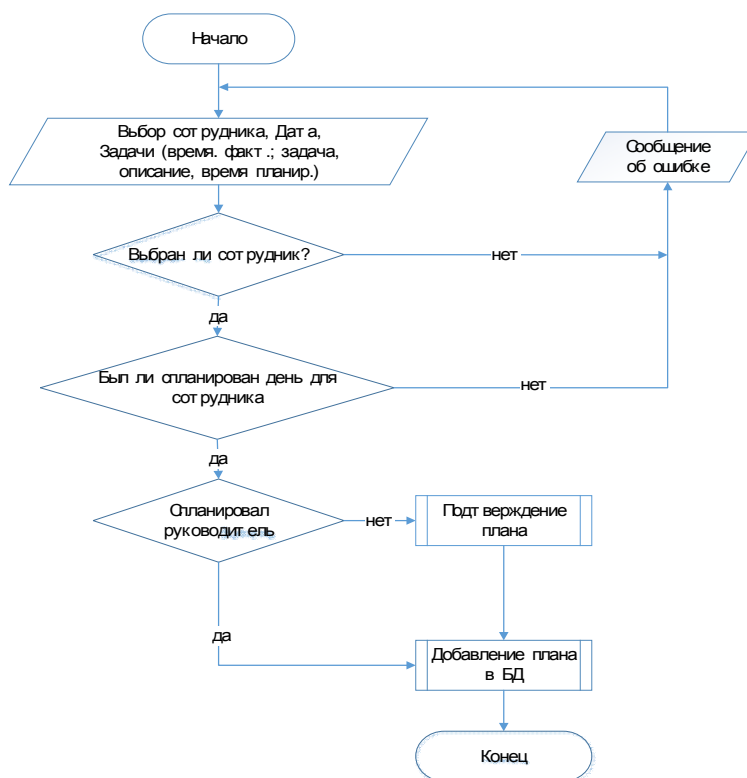


Рисунок 26 – Алгоритм планирования рабочего дня

В представленном алгоритме выше были показаны основные проверки на правильность ввода данных при добавлении плана в БД. При добавлении плана происходит проверка на то, был ли выбран сотрудник, для которого планируется рабочий день (если добавление плана производит сам руководитель, в ином случае данного поля не будет). После успешной проверки происходит дальнейший запрос к БД, если для данного сотрудника на выбранную дату уже был спланирован рабочий день, то сформируется сообщение об ошибке – «для данного сотрудника был уже спланирован день». После успешного прохождения всех проверок происходит добавление спланированного дня для сотрудника в БД, если планирование осуществлял руководитель, в ином случае спланированный день попадет на подтверждение руководителю, который в свою очередь либо подтвердит план с возможными внесенными изменениями, либо оставит план на потом.

Все возможные проверки на ошибки представлены в таблице 8.

Таблица 8 – Проверки на ошибки при добавлении плана в БД

Проверка на ошибку	Сообщение об ошибке	Часть кода
Повторение фактического времени	Повторение времени недопустимо	<pre>for (i = 0; i < countTask; i++) { for (j = (countTask-1); j > i; j--) if (arrTask[i][1] > arrTask[j][0] && arrTask[i][0] < arrTask[j][1] &&! =j) {isTimeRepeat = true; var numTimeRepeat = j;} if (isTimeRepeat) break;} </pre>
Правильный ввод диапазона времени	Неверно задан интервал времени	<pre>for (i = 0; i < countTask; i++) { if (arrTask[i][0] >= arrTask[i][1]) { isInterval = true; var numTimeRepeat = i;break;}} </pre>
Затронут ли обеденное время	Была спланирована задача на обеденный перерыв	<pre>for (i = 0; i < countTask; i++) { if(arrTask[i][0]<'13:00'&& arrTask[i][1]>'12:00') { isLunchTime = true; numTimeRepeat = i;break;}} </pre>
Заполнено ли одно из полей: задача или описание задачи	Одно из полей должно быть заполнено	<pre>for (i = 0; i < countTask; i++) { if(arrTask[i][2]=="" && arrTask[i][3]==")") { isNopeTask = true; numTimeRepeat = i;break;}} </pre>

5.2.4 Реализация подсистемы сохранения данных для отчетности

Данный функционал системы позволяет формировать таблицу отчетности спланированных работ в виде excel-файла. В данном файле будет

составлена таблица аналогична той, которая была сформирована на главной странице.

Алгоритм сохранения данных в excel-файл представлен на рис. 27.

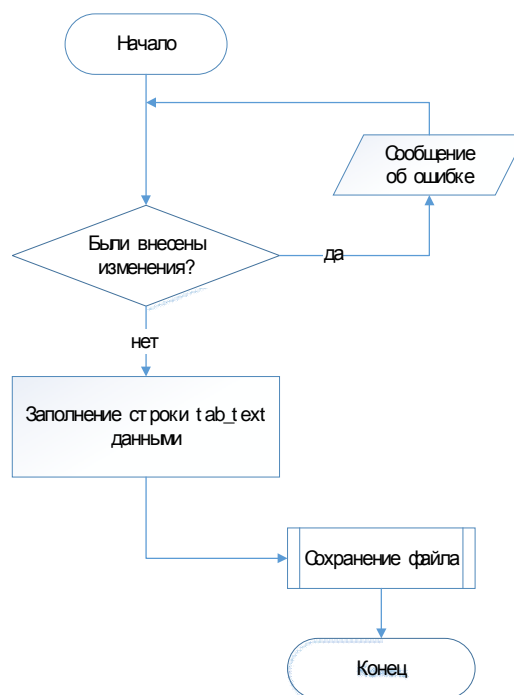


Рисунок 27 – Алгоритм сохранения файла для отчетности

При нажатии на кнопку «Экспорт в Excel» происходит проверка на то, были ли внесены изменения в таблицу до перезагрузки страницы, если проверка проходит успешно, то заполняется строка данными для дальнейшего переноса их в файл. А если были внесены изменения, то будет выведено сообщение «Были внесены изменения. Для того, чтобы сохранить документ перезагрузите страницу» и уже после перезагрузки страницы файл будет успешно создан (при повторном нажатии на соответствующую кнопку).

Часть программного кода представлена на рис. 28.


```

//функция сохранения таблицы в Excel
function printData() {
    if (isChange) {
        addUserMessage('Были внесены изменения. Для того, чтобы сохранить документ
        перезагрузите страницу <b>F5</b>', 'error');
        return;
    }
    tab = document.getElementById('table-print');
    for (j = 0; j < tab.rows.length; j++) {
        tab_text = tab_text + tab.rows[j].innerHTML + "</tr>";
    }
    tab_text = tab_text + "</table>";
    txtAreal.document.open("txt/html", "replace");
    txtAreal.document.write(tab_text);
    txtAreal.document.close();
    txtAreal.focus();
    sa = txtAreal.document.execCommand("SaveAs", true, "Отчет.xls");
    return (sa);
}

```

Рисунок 28– Часть программного кода сохранения файла для отчета

5.3 Описание запросов к сущностям базы данных

Для того чтобы иметь доступ к данным и делать запросы к сущностям, необходимо создавать подключение к СУБД на сервере, где хранится вся информация в таблицах. Функция подключения к СУБД представлена на рис. 29.

```

function connectDB () {
    $host = 'localhost';
    $database = 'gnewspon_VKR';
    $user = 'root';
    $password = '';
    global $mysqli;
    $mysqli = new mysqli($host, $user, $password, $database);
    if ($mysqli->connect_error) {
        printf("Connect failed: %s\n", $mysqli->connect_error);
        exit();
    }
    $mysqli->query( "SET CHARSET utf8" );
}

```

Рисунок 29 – Функция подключения к БД

В переменных \$host, \$database, \$user и \$password хранится информация о подключении к БД (имя хоста, название БД, логин и пароль соответственно). Для подключения создается новый класс mysqli в конструктор которого передаются параметры, указанные выше. Далее производится проверка на правильность подключения к БД, если данные были заданы правильно, то

проверка пройдет успешно. И появляется дальнейшая возможность работы со всеми данными. Работа с данными осуществляется при помощи SQLзапросов.

Рассмотрим основные запросы к сущностям, которые были использованы в реализации каждой из подсистем (таблица 9).

Таблица 9 – Основные SQLзапросы к сущностям

Подсистема	Запрос SQL	Описание
1	2	3
Авторизация пользователя	SELECT * FROM `users` WHERE `email` LIKE '\$email' AND `password` LIKE '\$password'	Происходит поиск сотрудника, данные которого совпадут с введенными пользователем при авторизации
Добавление сотрудника	INSERT INTO `users` ("first_name", "last_name", "middle_name", "email", "password", "id_leader") VALUES ('\$fName', '\$lName', '\$mName', '\$email', '\$psw', \$idLeader)	Производится добавление новой записи в таблицу «users» данными, которые были введены руководителем при добавлении сотрудника
Добавление плана	INSERT INTO `date_plan` ("date", "id_user") VALUES ('\$date', '\$idUser')	Добавление новой записи в таблицу «date_plan» данными – дата планирования и идентификатор сотрудника
	INSERT INTO `tasks` ("id_dateplan", "time_with", "time_before", "title",	Добавление записи в таблицу «tasks» данными, которые были заполнены

1	2	3
	"description", "plan_time") VALUES (\$data')	при планировании дня. \$data– массив данных.
Вывод спланированных задач	SELECT * FROM `date_plan`, `tasks` WHERE date_plan.id=tasks.id_dateplan ORDER BY date_plan.date DESC	Происходит выбор из двух таблиц «дата планирования» и «список задач»
Список сотрудников	SELECT * FROM `users` WHERE `id_leader`=\$idLeader ORDER BY "last_name" ASC	Вывод списка сотрудников в подчинении у руководителя

РУКОВОДСТВО ПОЛЬЗОВАТЕЛЯ ИНФОРМАЦИОННОЙ СИСТЕМЫ

В данном разделе будут описаны и проиллюстрированы окна разработанной ИС, а также основные моменты работы системы.

6.1 Авторизация пользователя

При первом запуске ИС пользователю будет предложено пройти авторизацию для дальнейшей работы с функционалом. Форма авторизации представлена на рис. 30.

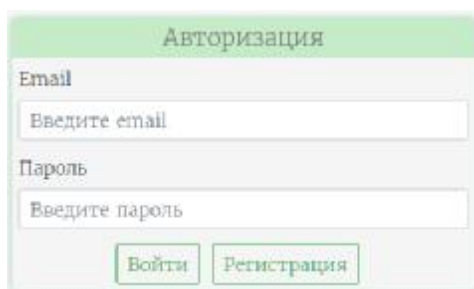
The image shows a web form titled "Авторизация" (Authorization) with a green header. It contains two input fields: "Email" with the placeholder text "Введите email" and "Пароль" (Password) with the placeholder text "Введите пароль". Below the fields are two buttons: "Войти" (Login) and "Регистрация" (Registration).

Рисунок 30 – Форма авторизации пользователя

Пользователь должен ввести свои данные (рис. 31) для входа в ИС. При этом если «email» или «пароль» будут введены неверно, то на экране появится соответствующее сообщение об этом (рис. 32 и рис. 33).

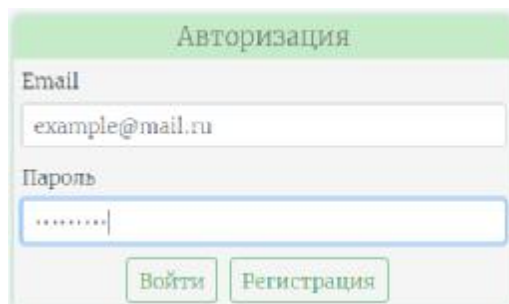
The image shows the same "Авторизация" form as in Figure 30, but with data entered. The "Email" field contains "example@mail.ru" and the "Пароль" field contains several dots representing a masked password. The "Войти" and "Регистрация" buttons are still visible.

Рисунок 31–Заполнение формы авторизации

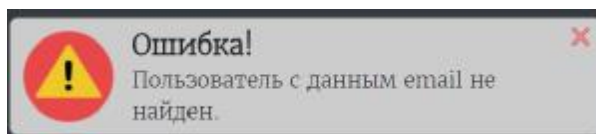


Рисунок 32 – Сообщение об ошибке при неверно введенной почты

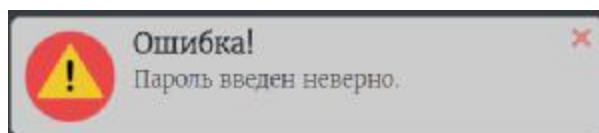


Рисунок 33 – Сообщение об ошибке при неверном пароле

При успешной авторизации предоставляется доступ к функционалу ИС и происходит перенаправление пользователя на главную страницу (п. 4.1). Для успешной авторизации в системе необходимо быть зарегистрированным в ИС и подтвержденным администратором. Регистрация пользователя представлена ниже (п.6.2).

6.2 Регистрация нового пользователя

Пользователю необходимо заполнить предложенную форму (рис. 34) для успешной регистрации в информационной системе. После нажатия на кнопку «Зарегистрироваться» происходит проверка на правильность введенных данных. Если введенная почта уже используется кем-то, то высветится на экране соответствующая ошибка (рис. 35).

Рисунок 34 – Заполнение формы добавления сотрудника

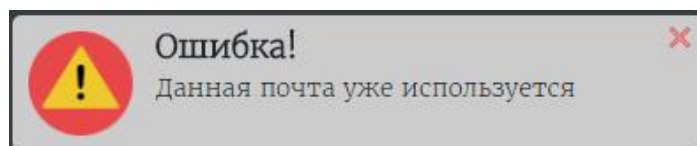


Рисунок 35 – Сообщение об ошибке при существующей почты

Так же происходит проверка на верность написания пароля, то есть совпадают ли введенные пароли. Если пароли не совпадают, то появляется соответствующее сообщение (рис. 36) об ошибке.

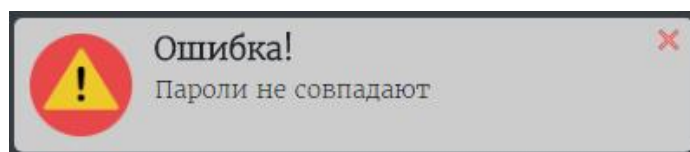


Рисунок 36 – Сообщение об ошибке при подтверждении пароля

При успешной регистрации, появляется сообщение (рис. 37) и для успешной авторизации в ИС необходимо ожидать подтверждения регистрации администратором.

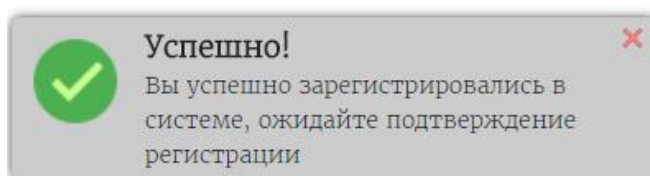


Рисунок 37 – Сообщение об успешном добавлении сотрудника

6.3 Планирование ремонтных работ

Планирование ремонтных работ доступно пользователям с правами доступа «руководитель» и «исполнитель». Предложенная форма для планирования представлена на рис. 36.

Рисунок 36 – Форма планирования работ

Так же в правой стороне рабочей области предложены блоки задач на выполнение (из файла или невыполненные ранее), которые можно мышкой перетащить в нужное поле (рис. 37).

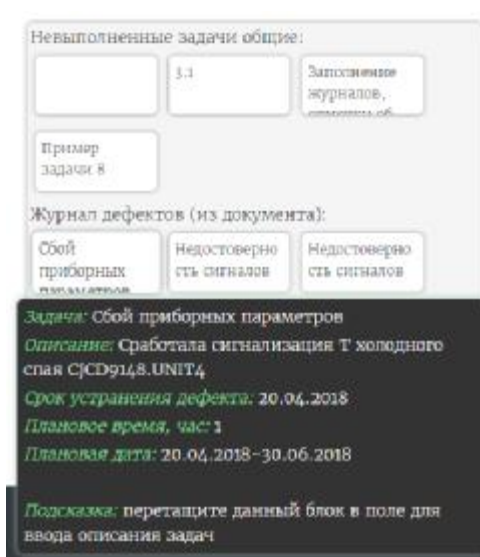


Рисунок 37 – Блоки предложенных задач на выполнение

Пример заполненной формы представлен на рис. 38.

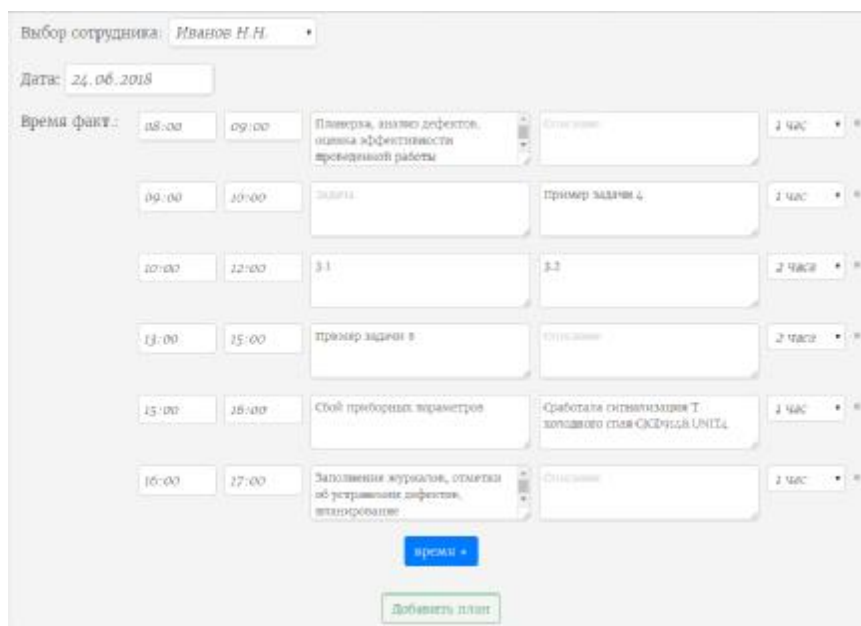


Рисунок 38 – Пример планирования ремонтных работ

Для добавления нового фактического времени необходимо нажать на кнопку «время+» (рис. 39).

Рисунок 39 – Кнопка добавления строки задачи

После того, как был спланирован рабочий день, необходимо нажать на кнопку «Добавить план», после чего данные будут обрабатываться и в случае возникновения ошибок будут появляться соответствующие оповещения. Возможные сообщения об ошибке: «не спланированы задачи» (рис. 40), «повторение времени не допустимо» (рис. 41), «неверно задан интервал времени» (рис. 42), «была спланирована задача на обеденный перерыв» (рис. 43), «одно из полей должно быть заполнено» (рис. 44).

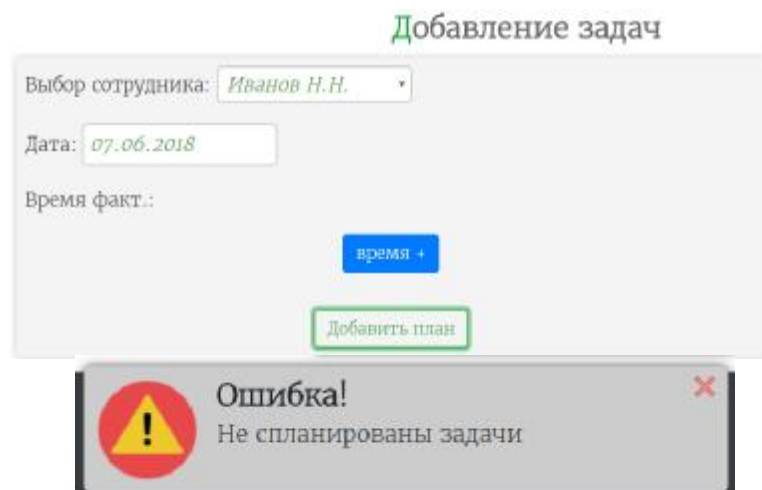


Рисунок 40 – Пример ошибки не спланированных задач

Добавление задач

Выбор сотрудника: *Иванов Н.Н.*

Дата: *07.06.2018*

Время факт.:

<i>08:00</i>	<i>09:00</i>	Планировка, анализ дефектов, оценка эффективности проведенной работы	Описание	1 час
<i>08:00</i>	<i>08:00</i>	Задача	Описание	1 час

время

Добавить план

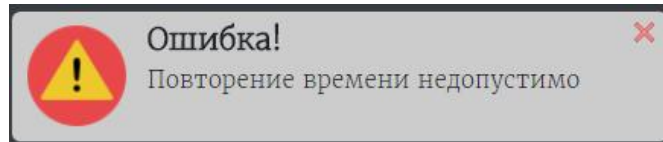


Рисунок 41 – Пример ошибки повторения времени

Добавление задач

Выбор сотрудника: *Иванов Н.Н.*

Дата: *07.06.2018*

Время факт.:

<i>08:00</i>	<i>09:00</i>	Планировка, анализ дефектов, оценка эффективности проведенной работы	Описание	1 час
<i>11:00</i>	<i>09:00</i>	Задача	Описание	1 час

время

Добавить план

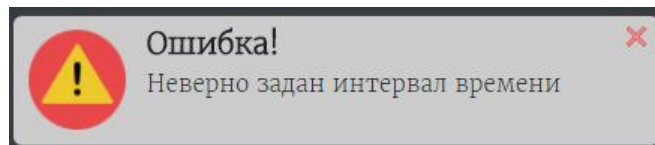


Рисунок 42 – Пример ошибки неверного интервала времени

Добавление задач

Выбор сотрудника: *Иванов И.И.*

Дата: 07.06.2018

Время факт.:

08:00	09:00	Планировка, анализ дефектов, оценка эффективности проводимой работы	Описание	1 час
12:00	14:00	Задача	Описание	1 час

[время]

[Добавить план]

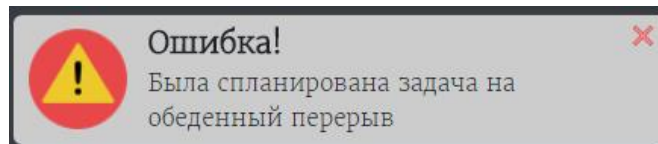


Рисунок 43 – Ошибка планирования на обеденный перерыв

Добавление задач

Выбор сотрудника: *Иванов И.И.*

Дата: 07.06.2018

Время факт.:

08:00	09:00	Планировка, анализ дефектов, оценка эффективности проводимой работы	Описание	1 час
09:00	10:00	Задача	Описание	1 час

[время]

[Добавить план]



Рисунок 44 – Ошибка на пустые поля задачи и описание задачи

А также наличие оповещения если задачи были спланированы не на весь рабочий день (рис. 45).

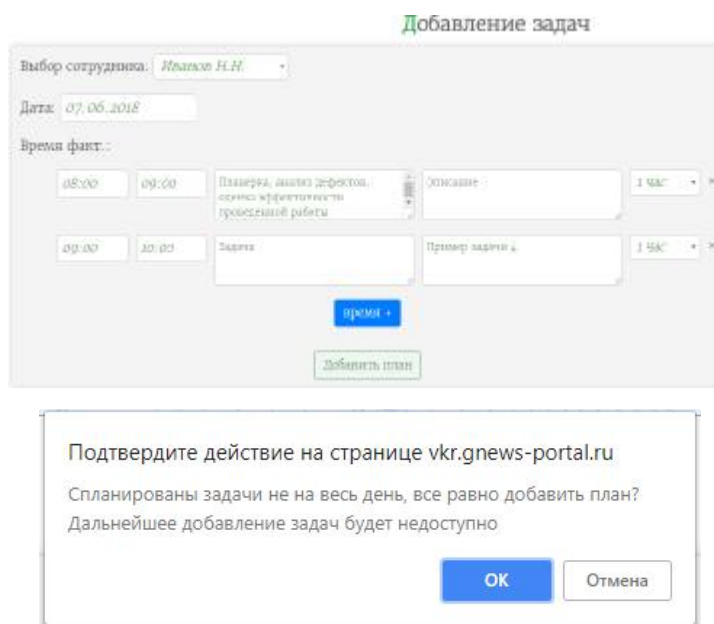


Рисунок 45 – Предупреждение о не полностью спланированном рабочем дне

После успешного добавления плана появляется сообщение (рис. 46) о том, что планирование завершилось успешно и добавлена в БД.

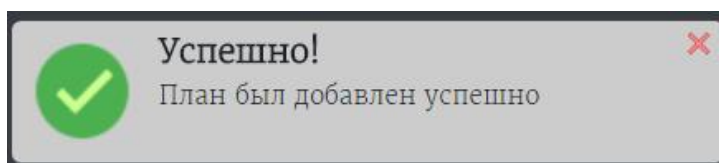


Рисунок 46 – Сообщение об успешном планировании

6.4 Фильтрация данных в таблице

Данный функционал является очень простым и примитивным, но в тоже время и очень необходимым. Фильтрация происходит посредством выбора необходимого значения из предложенных данных в форме. Пример выбора данных посредством задания фильтра представлен на рис. 47.

Выбор даты: ДД.ММ.ГГГГ

Выбор сотрудника: Иванов И.И.

Факт. времени с: ---:-- до: --:-- иное: -

Задача: -

Описание: -

Результат выполнения: Выполнено

Примечание: -

Допуск: -

Учет ремонта: -

Сортировать

Рисунок 47 – Форма фильтрации данных

После нажатия на кнопку «Сортировать» были получена таблица с данными, представленная на рис. 48.

Дата	ФИО	Время, факт.	Задача	Описание	Результат	Обоснование	Примечание	Допуск	Учет реп.	Часы учета / Плав. часы	Факт. час. с. доп.	Норм. допорог
22.05.2018	Иванов И.И.	0:00 - 0:00	Планировка, анализ дефектов, оценка эффективности		Выполнено		Подтверждено	да	1 / 1			
		16:00 - 17:00	Заполнение журналов, отметка об устранении		Выполнено		Не подтверждено	да	0 / 1			

Рисунок 48 – Таблица данных с использованием фильтра

6.5 Сохранения файла для отчетности

Сохранение файла в excel происходит путем нажатия на кнопку «Экспорт в Excel» (рис. 49). Если до перезагрузки страницы были внесены какие-либо изменения в таблицу, то появится сообщение об ошибке (рис. 50) и необходимо будет перезагрузить страницу для того, чтобы функция сохранения файла стала доступна.



Рисунок 49 – Кнопка сохранения файла

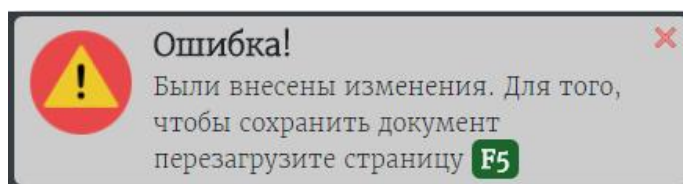


Рисунок 50 – Сообщение об ошибке сохранения файла

Если никаких ошибок не возникнет, то начнется скачивание excel файла на ПЭВМ. Полученная таблица в файле представлен на рис. 52.

Дата	ОНС	Время файла	Задача	Описание	Результат	Обоснование	Примечание	Доступ	Учет рем.	Часы учета / Показ.	Факт час. с доп.	План. доступ
22.09.2018	Новосиб. Г.Н.	0:00-0:00	Проверка наличия документов в архиве		Исполнено		Проверено	Д		0 / 0		
		0:00-0:00		Проверка данных в архиве	Не исполнено		Проверено	Д		0 / 0	0	0000
		0:00-0:00	Договорное задание, отклонено в архиве		Исполнено		Проверено	Д		0 / 0		
		0:00-0:00	Внесение в архив	Внесение в архив	Не исполнено		Проверено	Д		0 / 0		
		0:00-0:00	Задача		Не исполнено		Проверено	Д		0 / 0		
		0:00-0:00	Задача		Не исполнено		Проверено	Д		0 / 0		
		0:00-0:00	Задача		Не исполнено		Проверено	Д		0 / 0		

Рисунок 51 – Таблица планирования в ИС

1	2	3	4	5	6	7	8	9	10	11	12	
Дата	ОНС	Время файла	Задача	Описание	Результат	Обоснование	Примечание	Доступ	Учет рем.	Часы учета / Показ.	Факт час. с доп.	План. доступ
22.09.2018	Новосиб. Г.Н.	0:00-0:00	Проверка наличия документов в архиве		Исполнено		Проверено	Д		0 / 0		
		0:00-0:00	Проверка данных в архиве	Проверка данных в архиве	Не исполнено		Проверено	Д		0 / 0	0	0000
		0:00-0:00	Договорное задание, отклонено в архиве		Исполнено		Проверено	Д		0 / 0		
		0:00-0:00	Внесение в архив	Внесение в архив	Не исполнено		Проверено	Д		0 / 0		
		0:00-0:00	Задача		Не исполнено		Проверено	Д		0 / 0		
		0:00-0:00	Задача		Не исполнено		Проверено	Д		0 / 0		
		0:00-0:00	Задача		Не исполнено		Проверено	Д		0 / 0		
		0:00-0:00	Задача		Не исполнено		Проверено	Д		0 / 0		
		0:00-0:00	Задача		Не исполнено		Проверено	Д		0 / 0		

Рисунок 52 – Таблица планирования в excel файле

6.6 Подтверждение спланированных задач

Данный функционал доступен для пользователей с правами доступа руководитель. Выводится список спланированных задач исполнителем на подтверждение (рис. 53).

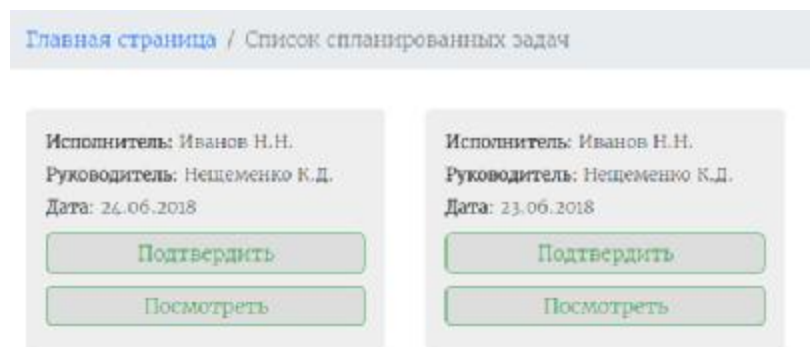


Рисунок 53 – Список спланированных задач

Данные задачи можно подтвердить, нажатием на кнопку «Подтвердить» или открыть на просмотр (рис. 54) и, если необходимо внести изменения и после добавить на исполнение.

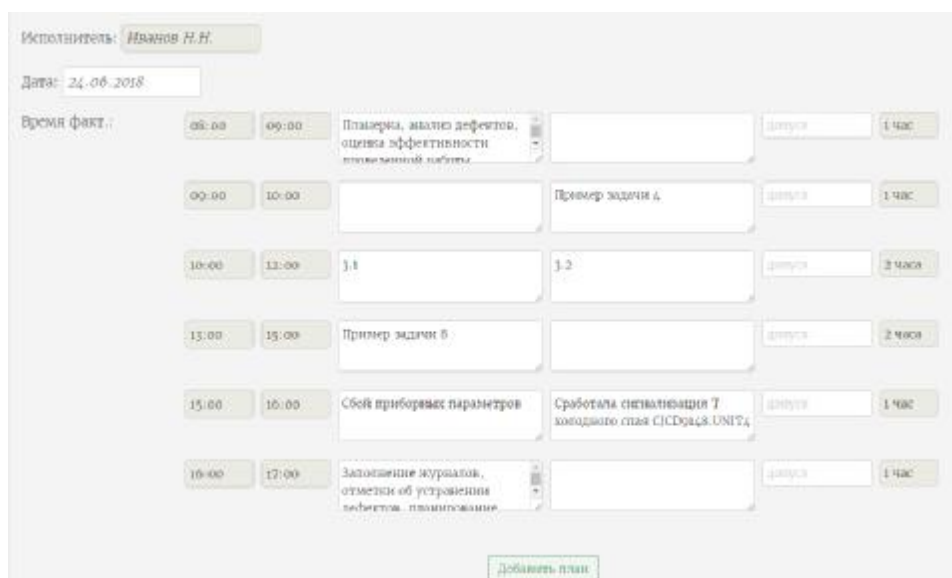


Рисунок 54 – Просмотр спланированных задач сотрудника

7 БЕЗОПАСНОСТЬ ЖИЗНЕДЕЯТЕЛЬНОСТИ

Целью выпускной квалификационной работы является разработка ИС. Рабочее место оператора ПК, на котором будет производиться работа с разработанной системой, должно соответствовать нормам безопасности жизнедеятельности.

Данный раздел позволит повысить производительность труда обслуживающего и ремонтного персонала, тем самым обеспечив эффективную работу оборудования.

7.1 Безопасность

Главным фактором, влияющим на производительность труда людей, работающих с ПЭВМ и ВДТ, являются комфортные и безопасные условия труда.

При выполнении работ на персональном компьютере (ПК) согласно ГОСТу 12.0.003-74 «ССБТ. Опасные и вредные производственные факторы. Классификация» [13] могут иметь место следующие факторы:

- повышенная температура поверхностей ПК;
- повышенная или пониженная температура воздуха рабочей зоны;
- повышенная или пониженная влажность воздуха;
- повышенный уровень электромагнитных излучений;
- зрительное напряжение;
- нервно-эмоциональные перегрузки.

Рассмотрим организацию рабочего места персонала с ПЭВМ. Общие требования, согласно СанПиН 2.2.2/2.4.1340-03[14], к организации рабочих мест:

- при размещении рабочих мест с ПЭВМ расстояние между рабочими столами с видеомониторами должно быть не менее 2,0м, а расстояние между боковыми поверхностями видеомониторов - не менее 1,2м;

- экран видеомонитора должен находиться от глаз пользователя на расстоянии 600-700 мм, но не ближе 500;

- конструкция рабочего стула (кресла) должна обеспечивать поддержание рациональной рабочей позы при работе на ПЭВМ;

- поверхность сиденья, спинки и других элементов стула (кресла) должна быть полумягкой, с нескользящим, слабо электризующимся и воздухопроницаемым покрытием, обеспечивающим легкую очистку от загрязнений.

Согласно СанПиН 2.2.2/2.4.1340-03 схема размещения оборудования на рабочем месте оператора показана на рис. 53.

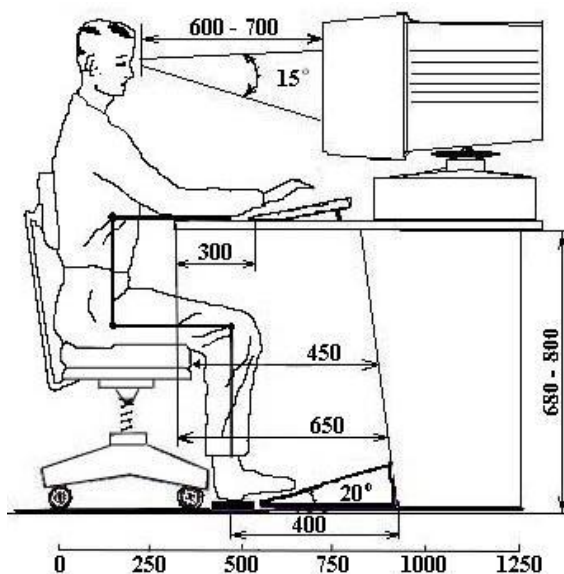


Рисунок 55 – Размещение оборудования на рабочем месте

Рабочее место пользователя ПЭВМ следует оборудовать подставкой для ног, имеющей ширину не менее 300 мм, глубину не менее 400 мм, регулировку по высоте в пределах до 150 мм и по углу наклона опорной поверхности подставки до 20°. Поверхность подставки должна быть рифленой и иметь по переднему краю бортик высотой 10 мм.

Клавиатуру следует располагать на поверхности стола на расстоянии 100-300 мм от края, обращенного к пользователю, или на специальной, регулируемой по высоте рабочей поверхности, отделенной от основной столешницы.

На рабочем месте при работе сПЭВМ параметры рабочего стола соответствуют нормам, стул регулируется по высоте в соответствии ростом работника, согласно СанПиН 2.2.2/2.4.1340-03.

В связи с постоянной работой оператора с ПЭВМ, невозможно не рассмотреть зрительную усталость, получившее название «компьютерный визуальный синдром». Изображение на экране не является привычным объектом наблюдения для человеческого глаза, поскольку на мониторе происходит мерцание кадров с определенной частотой.

При проектировании ИС для пользователя были соблюдены все требования цветокоррекции графических элементов. Требования к визуальным составляющим обязаны обеспечивать удобство работы пользователя, то есть минимизировать зрительную усталость и повысить надежность восприятия информации. Из-за особенностей зрительного восприятия, визуальные требования разделены на две группы. В первой группе представлены четыре параметра: яркость, освещенность, угловой размер знака, угол наблюдения. Параметры второй группы – неравномерность яркости, блики, мелькания, геометрические и нелинейные искажения и т.д.

Безопасность и комфортность зрительной работы с дисплеем ПЭВМ в большей степени зависит от контрастности изображения, то есть отношение яркости к яркости фона. Фоновая яркость образуется из-за диффузного отражения внешнего светового потока от люминофора трубки и зеркального отражения от стекла экрана.

7.2 Экологичность

Так как результатом разработки является создание информационной системы, то разработка может считаться экологически чистой, загрязнение окружающей среды и выбросов вредных веществ происходит не будет в связи с тем, что система является информационной и не управляет вредными производственными процессами.

7.3 Чрезвычайные ситуации

Пожар – это чрезвычайная ситуация, которая может возникнуть на любом объекте и нанести большой материальный вред, а также привести к гибели людей.[15] Пожар легче предупредить, чем потушить, поэтому при организации любого рабочего места необходимо предусмотреть мероприятия, которые устранят причины возникновения пожара.

Пожар в офисе, может привести к очень неблагоприятным последствиям (потеря ценной информации, порча имущества, гибель людей и т.д.), поэтому необходимо: выявить и устранить все причины возникновения пожара; разработать план мер по ликвидации пожара в здании; план эвакуации людей из здания.

Пожары возникают из-за причин неэлектрического и электрического характера. Рассмотрим причины возникновения пожаров электрического характера, к ним относят:

- короткое замыкание;
- перегрузки;
- искрение и электрические дуги;
- статическое электричество;
- и т.д.

К мерам предупреждения КЗ и перегрузок относят применение плавких предохранителей и специальных автоматов, включенных в цепь последовательно, а также правильный монтаж сетей, машин, аппаратов в соответствии с требованиями ПУЭ. Кроме того, следует правильно выбирать провода (рассчитывать их сечения) ЭУ, осуществлять профилактические осмотры, ремонты и испытания их.

Выбор системы сигнализации и средств пожаротушения.

Одним из основных факторов обеспечения пожарной безопасности на предприятиях является применение автоматических средств обнаружения пожаров, которые позволяют оповестить дежурный персонал о пожаре и месте его возникновения. Средствами пожарной сигнализации и оповещения

оборудуются производственные здания и помещения категорий А, Б и В, помещения с вычислительной техникой и дорогостоящей аппаратурой, а также объекты государственной важности.

В помещении с вычислительной техникой используются автоматические дымовые пожарные извещатели. Они реагируют на аэрозольные продукты горения. Автономные дымовые извещатели имеют собственный независимый источник питания, который необходимо менять один раз в год, и собственный оповещатель (сирену) с уровнем громкости 80 – 120 дБ. Контролируемая одним дымовым извещателем площадь в зависимости от типа может составлять до 80 м². Для повышения эффективности лучше устанавливать два извещателя.

В рассматриваемом помещении самыми вероятным источниками возгорания являются электроустановки (категория пожара Е - горение электроустановок), на этом будет основан выбор средств пожаротушения. При защите помещений ЭВМ следует учитывать специфику взаимодействия огнетушащих веществ с защищаемым оборудованием и материалами. Данные помещения следует оборудовать хладоновыми и углекислотными огнетушителями.

Огнетушители углекислотные относят к группе газовых. Ими возможно тушение загораний жидких горючих веществ и электрооборудования, находящегося под напряжением: ОУ-1 до 1000 В, а ОУ-3 и ОУ-5 до 10 кВ, также двуокись углерода не оказывает разрушающего действия на материалы.

Так как площадь помещения небольшая, достаточной будет установка 1 углекислотного огнетушителя 1-2 кг (ОУ-1 или ОУ-3).

ЗАКЛЮЧЕНИЕ

В ходе выполнения ВКР был рассмотрен объект автоматизации и рассмотрено существующее решение задачи. Так же было рассмотрено концептуальное проектирование системы, а именно обоснование выбора ПО, разбиение целой ИС на подсистемы и взаимодействие пользователя с системой. Спроектированы структуры базы данных и интерфейс пользователя информационной системы.

Была представлена программная реализация каждой из подсистем и рассмотрены основные запросы к сущностям БД. Приведено руководство пользователя ИС, для правильного взаимодействия сотрудника с системой.

Полученная система полностью соответствует поставленной задаче, достаточно полно выполняет все необходимые функции, а именно: хранение всех данных в едином архиве, планирование задач на рабочий день и автоматизированный процесс ведения учета.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

- 1 Амурская генерация. Акционерное общество «ДГК» [Электронный ресурс]. --Режим доступа: <https://www.dvgk.ru/page/2323>.- 04.04.2018
- 2 Balsamiq Products [Электронный ресурс]. --Режим доступа: <https://balsamiq.com/products/>.- 02.04.2018
- 3 WireframeSketcher Studio [Электронный ресурс]. --Режим доступа: <https://wireframesketcher.com>. -02.04.2018
- 4 1С: Предприятие 8. ERP Управление предприятием [Электронный ресурс]. --Режим доступа: <https://www.vdgb.ru/programmy-1s/1s-erp-upravlenie-predpriyatiem-2-0/>. -- 02.04.2018
- 5 SQLite Документация [Электронный ресурс]. -- Режим доступа: <https://www.sqlite.org/docs.html>. -- 08.05.2018
- 6 MySQL Документация [Электронный ресурс]. -- <https://dev.mysql.com/doc/>. -- 08.05.2018
- 7 PostgreSQL Документация [Электронный ресурс]. -- Режим доступа: <https://www.postgresql.org/docs/>. -- 08.05.2018
- 8 PHP [Электронный ресурс]. -- Режим доступа: <https://ru.wikipedia.org/wiki/PHP>. -- 07.06.2018
- 9 HTML [Электронный ресурс]. -- Режим доступа: <https://ru.wikipedia.org/wiki/HTML>. -- 07.06.2018
- 10 JavaScript [Электронный ресурс]. -- Режим доступа: <https://ru.wikipedia.org/wiki/JavaScript>. -- 07.06.2018
- 11 Nginx Документация [Электронный ресурс]. -- Режим доступа: <https://nginx.ru/ru/docs/>. -- 07.06.2018
- 12 Apache HTTP Server [Электронный ресурс]. -- Режим доступа: <https://httpd.apache.org/docs-project/>. -- 07.06.2018
- 13 ГОСТ 12.0.003-74 «ССБТ. Опасные и вредные производственные факторы. Классификация». ГОСТ Эксперт [Электронный ресурс]. -- Режим доступа: <http://gostexpert.ru/gost/gost-12.0.003-74#text>. -- 07.06.2018

14 СанПиН 2.2.2/2.4.1340-03. Гарант [Электронный ресурс]. – Режим доступа: <http://base.garant.ru/4179328/>. – 07.06.2018

15 Пожар [Электронный ресурс]. – Режим доступа: <https://ru.wikipedia.org/wiki/Пожар>. – 07.06.2018

16 ГОСТ 12.1.004-91 Система стандартов безопасности труда (ССБТ). Пожарная безопасность. Общие требования (с Изменением №1). – Взамен ГОСТ 12.1.004-85; введ. 01.07.1992. – Постановлением Государственного комитета СССР по управлению качеством продукции и стандартами от 14.06.91 N875

17 ГОСТ Р 50571.22-2000. «Электроустановки зданий. Часть 7. Требования к специальным электроустановкам. Раздел 707. Заземление оборудования обработки информации». – Введен впервые; введ. 01.01.2002; - Техн. комитетом по стандартизации ТК 337 «Электроустановки жилых и общественных домов»

ПРИЛОЖЕНИЕ А

Техническое задание

1 Общие сведения

Полное наименование: Создание автоматизированной системы планирования и учета работ по АСУ ТП на БТЭЦ.

Плановые сроки начала и окончания по созданию системы:

начало – 23 января 2018 года;

окончание – 29 июня 2018.

2 Назначение и цели создания автоматизированной системы

2.1 Назначение АС

ИС предназначена для:

- обеспечения руководства точной, достоверной и оперативной информацией о работе сотрудников;
- уменьшения материальных и трудовых затрат;
- хранения и обработки информации для отчетов;
- формирования журнала невыполненных задач с возможностью вывода на печать;
- отображения информации о поставленной задачи для исполнителя;
- регистрации аварийных событий;
- ведение оперативной документации и отчетности;
- анализ и планирование работ.

2.2 Цели создания ИС

- облегчение условий и повышение культуры труда персонала, за счет предоставляемого функционала системы;
- обеспечение персонала информацией (включая регистрацию событий, целей на выполнение задач, описания задачи) для анализа, оптимизации и планирования работы оборудования и его ремонтов;

Продолжение Приложения А Техническое задание

- обеспечения персонала достаточной, достоверной и своевременной информацией о ходе технологического процесса и состоянии оборудования для оперативного управления;
- улучшение интерфейса «человек-машина»;
- оперативный сбор, хранение, обработка и передача информации о состоянии поставленных задач персоналу;
- вывод отфильтрованной информации на печать;
- предоставление руководителю оперативную информацию о невыполненной поставленной задаче с обоснованием персонала.

3 Характеристики объекта автоматизации

3.1 Краткие сведения об объекте

Объект представляет собой программный интерфейс пользователя, связанный с базой данных, который расположен на локальном сервере на предприятии Благовещенской ТЭЦ.

3.2 Сведения об условиях эксплуатации объекта

Система планирования и учета ремонтных работ расположена на WEB-сервисе в открытом доступе для рабочего персонала, руководителя и оперативника. Добавлять план на исполнение может как исполнитель, так и руководитель.

4 Требования к системе

4.1 Требования к системе в целом

Информационная система должна состоять из следующих этапов проектирования:

- структурированной базы данных, со всеми связями между ее сущностями;
- пользовательский интерфейс, связанный с базой данных и полноценным функционалом.

4.1.1 Требования к структуре и функционированию системы

Продолжение Приложения А Техническое задание

Система должна иметь примитивный и понятный для персонала интерфейс, выполняющая поставленные для данной системы задачи.

АС должна включать основные прикладные подсистемы, выполняющие задачи автоматизации обмена информации и обработки задач, контроль выполненной работы персонала, планирование задач для персонала на день и ведение учета, а также вывод необходимой информации на печать.

4.1.2 Требования к численности и квалификации персонала системы и режиму его работы

Количество пользователей АСУ определяется текущими потребностями Благовещенской ТЭЦ.

Количество администраторов АСУ может быть определено по следующей методике: 1 администратор на 20-30 пользователей плюс 1 ведущий специалист или начальник отдела автоматизации.

Текущий контроль технического состояния оборудования АСУ ТП следует возложить на отдел автоматизации.

4.1.3 Показатели назначения

Целевое назначение системы должно сохраняться на протяжении всего срока эксплуатации Благовещенской ТЭЦ. Срок эксплуатации Благовещенской ТЭЦ определяется сроком устойчивой работы аппаратных средств вычислительных комплексов, своевременным проведением работ по замене (обновлению) аппаратных средств, по сопровождению программного обеспечения системы и его модернизации.

Прочие показатели назначения АСУ разрабатываются после проведения предпроектного обследования.

4.1.4 Требования к надежности

Показатели надежности:

Продолжение Приложения А Техническое задание

- время восстановления работоспособности прикладного ПО АСУ при любых сбоях и отказах не должно превышать одного рабочего дня, исключая случаи неисправности серверного оборудования;

- иные значения показателей надежности должны быть определены после проведения предпроектного обследования.

Требования к надежности

В АСУ должна быть обеспечена:

- корректная обработка сбоев электронно-механических устройств (например, принтеров) при выполнении функций, связанных с формированием твердых копий документов;

- возможность быстрой замены сбойного или вышедшего из строя активного накопителя на жестком диске (серверного оборудования АСУ) без остановки функционирования и потерь информации;

- возможность восстановления данных с внешнего накопителя после восстановления активного накопителя. Конкретный состав требований по восстановлению данных дополняется в соответствующими требованиями на подсистемы.

Осуществление разграничения прав доступа к системе.

Ведение журнала системы событий.

4.1.5 Требования безопасности

Аппаратное обеспечение данной системы должно соответствовать требованиям пожарной безопасности в производственных помещениях по ГОСТ 12.1.004-91.[16]

Аппаратное обеспечение системы должна быть заземлена в соответствии с требованиями ГОСТ Р 50571.22-2000.[17]

4.1.6 Требования к эргономике и технической эстетике

Требования к внешнему оформлению:

- реализация интерфейса пользователя;

Продолжение Приложения А Техническое задание

- приятная цветовая гамма для восприятия;
- читабельный текст.

Требования к диалогу с пользователем:

- интерфейс должен обеспечивать удобную навигацию;
- наличие помощи по добавлению задач.

4.1.7 Требования к эксплуатации, техническому обслуживанию, ремонту и хранению компонентов системы

Система должна обеспечивать непрерывный круглосуточный режим эксплуатации с учетом времени на техническое обслуживание.

В помещениях, предназначенных для эксплуатации системы, должны отсутствовать агрессивные среды, массовая концентрация пыли в воздухе должна быть не более $0,75 \text{ мг/м}^3$, электрическая составляющая электромагнитного поля помех не должна превышать $0,3 \text{ в/м}$ в диапазоне частот от $0,15$ до 300 МГц .

Напряжение питания сети должно быть $220\text{В} \pm 10$.

Нормальными климатическими условиями эксплуатации системы являются:

- температура окружающего воздуха $20 \pm 5 \text{ }^\circ\text{C}$;
- относительная влажность окружающего воздуха 60 ± 15 при температуре воздуха $20 \pm 5 \text{ }^\circ\text{C}$;
- атмосферное давление $101,3 \pm 4 \text{ Кпа}$.

Система должна сохранять работоспособность при воздействии следующих климатических факторов:

- температура окружающего воздуха от 10 до $35 \text{ }^\circ\text{C}$;
- относительная влажность воздуха от 40 до 80% при температуре $25 \text{ }^\circ\text{C}$.

4.1.8 Требования к защите информации от несанкционированного доступа

Для предотвращения доступа к информации третьим лицам, необходимо:

Продолжение Приложения А Техническое задание

- использовать ключи для шифрования данных;
- менять пароль от базы данных, хотя бы раз в неделю;
- подсистема управления доступом должна включать средства идентификации и контроль доступа пользователей к ресурсам;
- закрыть доступ к информации от лиц, не зарегистрированных в данной системе.

4.1.9 Требования по сохранности информации при авариях

Средствами обеспечения сохранности информации при авариях и сбоях в процессе эксплуатации являются:

- носители информации;
- создание резервных копий базы данных (ежедневно по расписанию);
- создание резервных копий программного обеспечения.

Для восстановления данных и программного обеспечения из резервной копии должны использоваться средства резервного копирования и архивирования.

АСУ должна обеспечивать возможность резервирования всех данных хранящиеся на серверах АСУ, а также возможность их восстановления.

4.1.10 Требования по стандартизации и унификации

Должен быть единый подход к решению однотипных задач, должны создаваться унифицированные объектно-ориентированные компоненты информационного, лингвистического, программного, технического и организационного обеспечения.

4.2 Требования к функциям (задачам), выполняемой системы

Функции подразделяются на информационные, управляющие и вспомогательные (сервисные).

В составе информационных функций входят следующие:

- сбора, первичной обработки и распределение информации;
- предоставление информации;

Продолжение Приложения А Техническое задание

- регистрация и архивирование событий;
- документирование информации.

В состав управляющих функций входят следующие:

- печать необходимой информации;
- вывод информации на экран отфильтрованной таблицы;
- назначение задач.

4.3 Требования к видам обеспечения

Серверное и клиентское программное обеспечение АСУ должно базироваться на согласованной с заказчиком и исполнителем распространенной промышленной сетевой операционной системе.

В настоящем ТЗ приведены специфические требования к лингвистическому, техническому и программному обеспечению.

4.3.1 Требования к лингвистическому обеспечению

Языки программирования

Разработка прикладного программного обеспечения должна вестись с использованием языков высокого уровня.

Языки взаимодействия пользователя и системы

Основным языком взаимодействия пользователей и системы является русский язык:

- взаимодействие пользователя с ПК должно осуществляться на русском языке;
- все документы и отчеты подсистемы готовятся и выводятся пользователю на русском языке;
- графический интерфейс пользователя подсистемы должен быть создан на русском языке.

Язык взаимодействия администраторов и системы

Продолжение Приложения А Техническое задание

Комплектование подсистемы ПО и документации на английском языке допускается только в том случае, если ПО и документация используется только администратором системы.

4.3.2 Требования к техническому обеспечению

Техническое обеспечение с указанными характеристиками должно быть достаточно для ввода подсистемы в опытную эксплуатацию на объекте автоматизации. При переводе подсистемы в промышленную эксплуатацию может потребоваться модернизация или замена технических средств на оборудование с другими характеристиками.

Для обеспечения регулярного резервного копирования необходимо предусмотреть возможность установки оборудования для выполнения резервного копирования информации на локальные накопители, либо возможность использование технических средств, позволяющих производить резервное копирование на выделенное сетевое устройство.

4.3.3 Требования к программному обеспечению

ПО системы должно строиться как распределённый программный комплекс, в котором программы и данные распределены между различными уровнями управления.

5 Состав и содержание работ по созданию (развитию) системы

Осуществление всего комплекса работ должно осуществляться в несколько очередей.

Этапы работ:

- формирование требований;
- эскизное проектирование;
- проектирование структуры базы данных;
- проектирование структуры интерфейса системы.

6 Порядок контроля и приемки системы

Продолжение Приложения А Техническое задание

Приемку работ должна осуществлять приемная комиссия, в состав которой включаются:

- представители заказчика;
- представители исполнителя.

Предварительные испытания заканчиваются подписание приемочной комиссией протокола испытания с указанием в нем перечня необходимых доработок программного обеспечения, программной и эксплуатационной документацией и сроков их выполнения.

7 Требования к составу и содержанию работ по подготовке объекта автоматизации к вводу системы в действие

В процессе создания подсистемы должен быть подготовлен и передан заказчику комплект документации в составе:

- проектная документация;
- программная и эксплуатационная документация;
- предложения по организации системно-технической поддержки функционирования подсистемы.

ПРИЛОЖЕНИЕ Б
Главная страница

Учет работы персонала Добро пожаловать, *Афанасов Л.С.* [Выйти](#)

[Добавить план](#)
[Настроить фильтр](#)
[Список сотрудников](#)

Дата	ФИО	Время факт	Задача	Описание	Результат	Обоснование	Примечание	Допуск	Учет рем.	Часы учета / План.	Факт час. с доп.	Норм. допуска
23.05.2018	Иванов Н.Н.	8:00-9:00	Планера, анализ дефектов, оценка		Выполнено		Подтверждено		да	1 / 1		
		9:00-11:00		Пример задачи 4	Не выполнено		Не подтверждено		да	2 / 2	2	ИНН
		16:00-17:00	Заполнение журналов, отметки об устранении		Выполнено		Не подтверждено		да	0 / 1		
		внеплан	Внеплановая задача	Внеплановое описание задачи	Не выполнено		Не подтверждено		да	1 / 1		
		офис. рем.	Офис. задача		Не выполнено		Не подтверждено		да	1 / 1		
		учет часов								5		

[Экспорт в Excel](#)

Рисунок 56 – Вид главной страницы

ПРИЛОЖЕНИЕ В
Страница планирования задач

Учет работы персонала Добро пожаловать, *Афанасов Л.С.* [Выйти](#)

[Главная страница](#) / [Добавление плана](#)

Добавление задач

Выбор сотрудника:

Дата:

Время факт.:

<input type="text" value="08:00"/>	<input type="text" value="09:00"/>	<input type="text" value="Планерка, анализ дефектов, оценка эффективности проведенной работы"/>	<input type="text" value="Описание"/>	<input type="text" value="1 час"/>	×
<input type="text" value="08:00"/>	<input type="text" value="09:00"/>	<input type="text" value="Задача"/>	<input type="text" value="Описание"/>	<input type="text" value="1 час"/>	×

[время +](#)

[Добавить план](#)

Задачи невыполненные ранее:

Список задач (из документа):

<input type="text" value="Пример задачи 1"/>	<input type="text" value="Пример задачи 2"/>	<input type="text" value="Пример задачи 3"/>
<input type="text" value="Пример задачи 4"/>	<input type="text" value="Пример задачи 5"/>	<input type="text" value="Пример задачи 6"/>
<input type="text" value="Пример задачи 7"/>	<input type="text" value="Пример задачи 8"/>	

Рисунок 57 – Вид страницы добавления плана

ПРИЛОЖЕНИЕ Г
Страница списка сотрудников

№	Фамилия	Имя	Отчество	Почта	Последнее планирование
1	Иванов	Николай	Николаевич	inn@mail.ru	22.05.2018
2	Нещеменко	Константин	Дмитриевич	qwewqe@mail.ru	-

Рисунок 58 – Вид страницы «Список сотрудников»

ПРИЛОЖЕНИЕ Д Листинг программы

Главная страница:

```
<?php require_once "session.php" ?>
<!DOCTYPE html>
<html lang="ru">
<head>
    <?php
        $title = "ВКР";
        require_once "block/head.php";
        $allUser = getUser(null,null);
    ?>
</head>
<body>
    <!--Шапка -->
    <?php require_once "block/header.php" ?>

    <!--Контент -->
    <div class="container-fluid bg-light">
        <!-- Кнопки и панель пользователя -->
        <div class="row">
            <!-- Кнопки управления -->
            <div class="col p-3">
                <?php if (isset($_SESSION['logged_user'])) : ?><!-- Изменить! "!" -->
                    <? if ($_SESSION['logged_user'][USER_ID_GROUP] < 3) :?>
                        <button type="button" class="btn btn-outline-success"
id="addPlan" onclick="location.href = '/add-plan.php';">Добавитьплан</button>
                    <? endif; ?>
                        <button type="button" class="btn btn-outline-success"
id="addFilter">Настроитьфильтр</button>
                    <? if ($_SESSION['logged_user'][USER_ID_GROUP] == 1) : ?>
                        <button type="button" class="btn btn-outline-success"
onclick="location.href = '/listCo-Worker.php';">Списоксотрудников</button>
                    <? endif; ?>
                <?php endif; ?>
            </div>
            <!-- Авторизация/регистрация -->
            <?php if(!isset($_SESSION['logged_user'])) : ?>
                <div class="col-3 m-3 authAndReg">
                    <?php require_once "block/authAndReg.php" ?>
                </div>
            </?>
        </div>
    </div>
</body>
</html>
```

Продолжение Приложения Д Листинг программы

```
</div>
<?php endif; ?>
</div>
<!-- Выбор фильтра -->
<div class="row">
    <div class="col">
        <?php require_once "block/formFilter.php" ?>
    </div>
</div>
<!-- Таблица -->
<div class="row">
    <div class="col">
        <?php
            if (isset($_SESSION['logged_user'])) {
                require_once "block/inputData.php";
                if ($_SESSION['logged_user'][USER_ID_GROUP] != 3 &&
count($listDatePlan) > 0) {
                    ?>
                        <div class="row justify-content-end p-3 mb-2">
                            <button type="button" class="btn btn-outline-success"
onclick="printData()">Экспорт в Excel</button><!--            изменить блок,            в нем без инпут,
текст старе сделать вывод таблицы!!! -->
                        </div>
                    <?php }
                    }
                    else
                        echo '<div class="p-2 m-3 bg-danger text-white"><i class="fs-
16">Информация недоступна, необходимо пройти авторизацию</a></i></div>';
                    ?>
                </div>
            </div>
        </div>
<!-- Подвал -->
<?php require_once "block/footer.php" ?>

<!-- Скрипты js -->
<script>
    // Развернуть/свернуть форму фильтра
    $('#addFilter').click(function() {
```

Продолжение Приложения Д Листинг программы

```
if ($('#formFilter')[0].style.display == 'block')
    $('#formFilter').css('display', 'none');
else
    $('#formFilter').css('display', 'block');
});

// Изменения/добавления данных задач
var isChange = false;
function changeData(objDiv, taskID) {
    var nameID = objDiv.id;
    var val = objDiv.value;

    if (taskID && (val != 0 || val == "")) {
        //console.dir(objDiv.parentElement.parentElement.children);

        //POST запрос
        $.ajax({
            url: 'ajax/changeTask.php',
            type: 'POST',
            cache: false,
            data: ({val: val, name: nameID, id: taskID}),
            dataType: 'html',
            success: function(data) {
                if (data == 'success') {

addUserMessage('Изменения были успешно сохранены', data);
                    isChange = true;
                }
                else {
                    addUserMessage(data, 'error');
                }
            }
        });
    }
}

function addDataTask(objDiv, dateID, type) {
    var nameID = objDiv.id;
    var val = objDiv.value;
```

Продолжение Приложения Д Листинг программы

```
if (dateID&& (val != 0 || val != "") && !isChange) {
    //POST запрос
    $.ajax({
        url: 'ajax/changeTask.php',
        type: 'POST',
        cache: false,
        data: ({val: val, name: nameID, id: dateID, type:type}),
        dataType: 'html',
        success: function(data) {
            if (data == 'success') {
                addUserMessage('Данные были успешно добавлены',
data);
                isChange = true;
            }
            else {
                addUserMessage(data, 'error');
            }
        }
    });
}
else if (isChange)
    addUserMessage('Были внесены изменения,
пожалуйста перезагрузите страницу <b>F5</b>', 'error');
else
    addUserMessage('Пожалуйста введите данные', 'error');
}

//функция сохранения таблицы в Excel
function printData() {
    if (isChange) {
        addUserMessage('Были внесены изменения. Для того, чтобы сохранить
документ перезагрузите страницу <b>F5</b>', 'error');
        return;
    }

    var tab_text = "<table border='2px' style='font-size:16px'><tr bgcolor='#212529'
style='color:white;height:40px'>";
    var j = 0;
```

Продолжение Приложения Д Листинг программы

```
tab = document.getElementById('table-print');

for (j = 0; j <tab.rows.length; j++) {
tab_text = tab_text + tab.rows[j].innerHTML + "</tr>";
}

tab_text = tab_text + "</table>";

var ua = window.navigator.userAgent;
var msie = ua.indexOf("MSIE ");
if (msie > 0 || !!navigator.userAgent.match(/Trident.*rv\:11\./)) // If Internet Explorer
{
txtArea1.document.open("txt/html", "replace");
txtArea1.document.write(tab_text);
txtArea1.document.close();
    txtArea1.focus();
sa = txtArea1.document.execCommand("SaveAs", true, "Отчет.xls");
} else //other browser not tested on IE 11
sa = window.open('data:application/vnd.ms-excel,' + encodeURIComponent(tab_text));

return (sa);

}

</script>

</body>
</html>
```

Страница списка сотрудников:

```
<?php
require_once "session.php";
if(!isset($_SESSION['logged_user']) || $_SESSION['logged_user'][USER_ID_GROUP]==3 ):
?>
<script>document.location.href="/"</script>
<? endif;?>
<!DOCTYPE html>
<html lang="ru">
```

Продолжение Приложения Д Листинг программы

```
<head>
  <?php
    $title = "Список сотрудников | ВКР";
    require_once "block/head.php";
  ?>
</head>
<body>
  <!--Шапка -->
  <?php require_once "block/header.php" ?>

  <!--Контент -->
  <div class="container-fluid">
    <!-- Миникарта сайта -->
    <div class="row">
      <div class="map-site-bg">
        <nav aria-label="breadcrumb">
          <ol class="breadcrumb">
            <li class="breadcrumb-item"><a href="/" class="fs-
16">Главная</a></li>
            <li class="breadcrumb-item active fs-16" aria-
current="page"><?if($_GET['id'] || $_SESSION['logged_user'][USER_ID_GROUP]==2) echo
'Эффективность работы сотрудника'; else echo 'Список сотрудников'; ?></li>
          </ol>
        </nav>
      </div>
    </div>

    <!-- Кнопка добавления сотрудника -->
    <? if ($_SESSION['logged_user'][USER_ID_GROUP]==1): ?>
    <div class="row">
      <div class="col p-3">
        <? if (!$_GET['id']): ?>
        <button type="button" class="btn btn-outline-success" onclick="location.href =
'/reg.php'">Зарегистрировать</button>
        <? elseif ($_SESSION['logged_user'][USER_ID_GROUP]==1): ?>
        <button type="button" class="btn btn-outline-success" onclick="location.href =
'/listCo-worker.php'">Список сотрудников</button>
        <? endif; ?>
      </div>
    </div>
  </div>
</body>
```


Продолжение Приложения Д Листинг программы

```

</div>
<? endif; ?>
<!--Списоксотрудников -->
<? if ($_SESSION['logged_user'][USER_ID_GROUP]==1 && !$_GET['id']): ?>
<div class="row">
    <div class="col">
        <table class="table table-striped">
            <thead class="thead-dark">
                <tr>
                    <th scope="col">№</th>
                    <th scope="col">Фамилия</th>
                    <th scope="col">Имя</th>
                    <th scope="col">Отчество</th>
                    <th scope="col">Почта</th>
                    <th scope="col">Последнее планирование</th>
                </tr>
            </thead>
            <tbody>
                <?
                    $coWorker
                    =
getUser(null,$_SESSION['logged_user'][USER_ID]);
                    for ($i=0; $i<count($coWorker); $i++) :
                ?>
                <tr>
                    <th><?=( $i+1 )?></th>
                    <td><?=$coWorker[$i][USER_LAST_NAME]?></td>
                    <td><?=$coWorker[$i][USER_FIRST_NAME]?></td>
                    <td><?=$coWorker[$i][USER_MIDDLE_NAME]?></td>
                    <td><?=$coWorker[$i][USER_EMAIL]?></td>
                    <td><?=formatDate(getDatePlan(false,"WHERE
".DATE_PLAN_ID_USER."=".$coWorker[$i][USER_ID])[0][DATE_PLAN_DATE])?></td>
                </tr>
                <? endfor; ?>
            </tbody>
        </table>
    </div>
</div>
<? endif; ?>
</div>

```

Продолжение Приложения Д Листинг программы

```
<!--Подвал -->
<?php require_once "block/footer.php" ?>
</body>
</html>
```

Страница планирования:

```
<?php require_once "session.php" ?>
<!DOCTYPE html>
<html lang="ru">
<head>
  <?php
    $allUser = getUser(null,null);
    $title = "Добавление плана | ВКР";
    require_once "block/head.php";
  ?>
</head>
<body>
  <!--Шапка -->
  <?php require_once "block/header.php"; if (!isset($_SESSION['logged_user']) ||
$_SESSION['logged_user'][USER_ID_GROUP] > 2) :?>
    <script>document.location.href="/"</script>
  <?php endif; ?>
  <!--Контент -->
  <div class="container-fluid">
    <div class="row">
      <!-- Миникарта сайта -->
      <div class="map-site-bg">
        <nav aria-label="breadcrumb">
          <ol class="breadcrumb">
            <li class="breadcrumb-item"><a href="/">Главная страница</a></li>
            <li class="breadcrumb-item active" aria-
current="page">Добавление плана</li>
          </ol>
        </nav>
      </div>
      <div class="reg-user">
        <div class="titleReg">Добавление задач</div>
      </div>
```

Продолжение Приложения Д Листинг программы

```
<!-- Форма планирования -->
<div class="leftCol">
  <form id="formdata" method="post" class="formPlan" onsubmit="return false"
enctype="multipart/form-data">
  <!--Выборсотрудника -->
  <?php if($_SESSION['logged_user'][USER_ID_GROUP] == 1): ?>
    <div class="form-plan-group">
      <label>Выборсотрудника:
        <select name="co-worker" id="co-worker" required>
          <option value="0"></option>
        <?php
          for ($i = 0; $i < count($allUser); $i++)
            if ($allUser[$i][USER_ID_GROUP]
== 2) echo '<option value="'. $allUser[$i][USER_ID]."'>'.getFIO($allUser[$i]).'</option>';
        ?>
      </select>
    </label>
  </div>
  <?php endif; ?>
  <!--Дата -->
  <div class="form-plan-group">
    <label>Дата: <input type="date" name="date" id="date"
value="<?=date('Y-m-d');?>" required></label>
  </div>
  <!--Времяфакт -->
  <div class="form-plan-group" id="addTask">
    <label>Время факт.: </label>
    <!-- Шаблон времени -->
    <script>$(document).ready(function(){ addTask();});</script>
  </div>
  <!--Добавитьвремя -->
  <div class="bttn-center">
    <button type="submit" class="btn btn-primary" id=""
onclick="addTask(); return false;">время +</button>
  </div>
  <!-- Добавить план в бд -->
  <div class="bttn-center" style="margin-top:35px;">
    <button type="submit" class="btn btn-outline-success"
id="bttnAddPlan" onclick="return false;">Добавитьплан</button>
```

Продолжение Приложения Д Листинг программы

```
</div>
</form>
</div>

<!-- Задачи, не выполненные ранее -->
<div class="rightCol">
<?
    # подключаем библиотеку
    require_once "block/PHPExcel.php";
    # Указываем путь до файла .xlsx
    $File = "tasks.xlsx";
    $Excel = PHPExcel_IOFactory::load($File);
    $notDonePlan = getNotDonePlan();
    if ($notDonePlan || $Excel) :
?>
    <div class="not-done-tasks" id="done-plan">
        <span class="list-plan">Задачи невыполненные ранее:</span>
        <?
            for ($i=0; $i<count($notDonePlan); $i++):
                if ($notDonePlan[$i][NOT_DONE_TASK_DATE] <
date('Y-m-d')) :
                    ?>
                        <div class="dragndrop" ondragstart="dragStart(event)"
draggable="true"
                        id="done-plan-?=$notDonePlan[$i][NOT_DONE_TASK_ID]?"
onmouseover="showMessage('<i>Полный текст:</i><?=$notDonePlan[$i][NOT_DONE_TASK_TITLE]?<br><i
>Подсказка:</i>перетащите данный блок в поле для ввода описания задач',this)" onmouseout="closeMessage()"
                        <?=$notDonePlan[$i][NOT_DONE_TASK_TITLE]?>
                        </div>
                    <?
                        endif;
                    endfor;
                ?>
            <span class="list-plan">Список задач (из документа):</span>
            <!-- Задачи, не выполненные ранее из файла Excel -->
            <?
                # С какой строки начинаются данные
                $Start = 2;
                $Res = array();
```

Продолжение Приложения Д Листинг программы

```
        for ($i= $Start; $i <= 1000; $i++) {
            $Row = new stdClass();

            $Row->id = $i;
            $Row->title      =      $Excel->getActiveSheet()-
>getCell('A'.$i )->getValue();
            // $Row->plan_time = $Excel->getActiveSheet()-
>getCell('B'.$i )->getValue();

            if($Row->title == null) continue;
            $Res[] = $Row;
        }
    ?>
    <? for($i = 0; $i < count($Res); $i++): ?>
        <div class="dragndrop" ondragstart="dragStart(event)"
draggable="true" id="done-plan-file-<?=$Res[$i]->id?">
onmouseover="showMessage('<i>Полный текст:</i><?=$Res[$i]-
>title?><br><i>Подсказка:</i>перетащите данный блок в поле для ввода описания задач',this)"
onmouseout="closeMessage()">
            <?=$Res[$i]->title?>
        </div>
    <? endfor; ?>
</div>
<? endif; ?>
</div>
</div>
<!-- Подвал -->
<?php require_once "block/footer.php" ?>
<!-- Скрипты js -->
<script>
    //Покрутка невыполненных задач за скролингом страницы
    $(function() {
        var offsetTop = $('<i>.not-done-tasks</i>')[0].offsetTop;
        $(window).scroll(function() {
            var height = $('<i>.not-done-tasks</i>')[0].offsetHeight;
            var cscrollTop = $(this).scrollTop();
            if( cscrollTop > offsetTop) {
                $('<i>.not-done-tasks</i>').css('top', (cscrollTop-offsetTop+10) +'px');
```

```
} else {
```

Продолжение Приложения Д Листинг программы

```
    $('.not-done-tasks').css('top', 0);
  }
});
//Drag and drop
function dragStart(event) {
    event.dataTransfer.setData("Text",event.target.innerText);
    event.dataTransfer.setData("IdPlan",event.target.id);
}
function allowDrop(event) {
    event.preventDefault();
}
function drop(event) {
    var data = event.dataTransfer.getData("Text"); // позволяет получить данные.
    event.target.value += data; // добавляем перетаскиваемый элемент

    var dataId = event.dataTransfer.getData("IdPlan");
    document.getElementById(dataId).remove();
    var list = document.getElementsByClassName('dragndrop');
    if (list.length == 0) {
        document.getElementById('done-plan').remove();
        document.getElementsByClassName('leftCol')[0].style.width = '100%';
        document.getElementsByClassName('leftCol')[0].style.margin = '1% 13%';
    }
}
//Добавление/удаление времени
var numberTask = 0;
function addTask() {
    var task = 'task-' + numberTask;
    var phpSelect;
    var templateTask;
    var lastTime = "";
    while (document.getElementById(task)) {
        numberTask++;
        task = 'task-' + numberTask;
    }
    if (numberTask > 10) return;
    switch(numberTask) {
```

case 0:

Продолжение Приложения Д Листинг программы

```
phpSelect = '<input type="time" min="08:00" max="17:00"
class="inputTime" id="timeTask_with" value="08:00"
onmouseover="showMessage(\<i>Подсказка:</i>Введите фактическое время начала выполнения задачи\',this)"
onmouseout="closeMessage()"><input type="time" min="08:00" max="17:00" class="inputTime"
id="timeTask_before" value="09:00"
onmouseover="showMessage(\<i>Подсказка:</i>Введите фактическое время окончания выполнения задачи\',this
)" onmouseout="closeMessage()">;
```

```
templateTask = '<div class="timeFuct" id="'+ task +"'> + phpSelect +
'<textarea name="titleTask" id="titleTask">Планерка, анализ дефектов,
оценка эффективности проведенной работы</textarea><textarea name="descriptionTask" id="descriptionTask"
placeholder="Описание" ondrop="drop(event)" ondragover="allowDrop(event)"></textarea><select
class="inputTime" name="plan-time" id="plan-time"
onmouseover="showMessage(\<i>Подсказка:</i>Выберите планируемых количество часов на выполнение задачи\'
,this)" onmouseout="closeMessage()" onmousedown="closeMessage()"><option value="1" selected>1
час</option><option value="2">2 часа</option><option value="3">3 часа</option><option value="4">4
часа</option><option value="5">5 часов</option><option value="6">6 часов</option><option value="7">7
часов</option><option value="8">8 часов</option></select><button type="button" class="close" aria-
label="Close" onclick="deleteTask(\'+ task + \')"><span aria-hidden="true">&times;</span></button></div>;
```

break;

case 7:

```
phpSelect = '<input type="time" min="08:00" max="17:00"
class="inputTime" id="timeTask_with" value="16:00"
onmouseover="showMessage(\<i>Подсказка:</i>Введите фактическое время начала выполнения задачи\',this)"
onmouseout="closeMessage()"><input type="time" min="08:00" max="17:00" class="inputTime"
id="timeTask_before" value="17:00"
onmouseover="showMessage(\<i>Подсказка:</i>Введите фактическое время окончания выполнения задачи\',this
)" onmouseout="closeMessage()">;
```

```
templateTask = '<div class="timeFuct" id="'+ task +"'> + phpSelect +
'<textarea name="titleTask" id="titleTask">Заполнение журналов, отметки об устранении дефектов,
планирование</textarea><textarea name="descriptionTask" id="descriptionTask" placeholder="Описание"
ondrop="drop(event)" ondragover="allowDrop(event)"></textarea><select name="plan-time" class="inputTime"
id="plan-time"
onmouseover="showMessage(\<i>Подсказка:</i>Выберите планируемых количество часов на выполнение задачи\'
,this)" onmouseout="closeMessage()" onmousedown="closeMessage()"><option value="1" selected>1
час</option><option value="2">2 часа</option><option value="3">3 часа</option><option value="4">4
часа</option><option value="5">5 часов</option><option value="6">6 часов</option><option value="7">7
часов</option><option value="8">8 часов</option></select><button type="button" class="close" aria-
label="Close" onclick="deleteTask(\'+ task + \')"><span aria-hidden="true">&times;</span></button></div>;
```

```
break;
```

```
default:
```

Продолжение Приложения Д Листинг программы

```
phpSelect = '<input type="time" min="08:00" max="17:00"
class="inputTime" id="timeTask_with" value="08:00"
onmouseover="showMessage('\<i>Подсказка:</i>Введите фактическое время начала выполнения задачи\',this)"
onmouseout="closeMessage()"'><input type="time" min="08:00" max="17:00" class="inputTime"
id="timeTask_before" value="09:00"
onmouseover="showMessage('\<i>Подсказка:</i>Введите фактическое время окончания выполнения задачи\',this
)" onmouseout="closeMessage()"'>;

templateTask = '<div class="timeFuct" id="'+ task +'>' + phpSelect +
'<textarea name="titleTask" id="titleTask" placeholder="Задача"></textarea><textarea name="descriptionTask"
id="descriptionTask" placeholder="Описание" ondrop="drop(event)"
ondragover="allowDrop(event)"></textarea><select name="plan-time" class="inputTime" id="plan-time"
onmouseover="showMessage('\<i>Подсказка:</i>Выберите планируемых количество часов на выполнение задачи\'
,this)" onmouseout="closeMessage()" onmousedown="closeMessage()"'><option value="1" selected>1
час</option><option value="2">2 часа</option><option value="3">3 часа</option><option value="4">4
часа</option><option value="5">5 часов</option><option value="6">6 часов</option><option value="7">7
часов</option><option value="8">8 часов</option></select><button type="button" class="close" aria-
label="Close" onclick="deleteTask('\'+ task +'\')"><span aria-hidden="true">&times;</span></button></div>';

break;
}

$('#addTask').append(templateTask);
}

function deleteTask(numberDeleteTask) {
    $('#'+ numberDeleteTask).remove();
    numberTask = 0;
}

//Проверка заполненных данных и POST запрос на добавление плана
$('#btnAddPlan').click(function() {
    var isTimeRepeat = false;
    var isInterval = false;
    var isNotFullPlanned = false;
    var isNopeTask = false;
    var minTime = '17:00';
    var maxTime = '08:00';
    var isLunchTime = false;
    var arrTask = [];
    var contentTask = document.getElementsByClassName('timeFuct');
```



```

var countTask = contentTask.length;
//Проверка спланирован ли день
if (countTask == 0) {
    Продолжение Приложения Д
    Листинг программы
    addUserMessage('Неспланированызадачи', 'error');
    return;
}
for (i = 0; i < countTask; i++) {
    arrTask[i] = [];
    arrTask[i][0] = contentTask[i].children[0].value;
    arrTask[i][1] = contentTask[i].children[1].value;
    arrTask[i][2] = contentTask[i].children[2].value;
    arrTask[i][3] = contentTask[i].children[3].value;
    arrTask[i][4] = contentTask[i].children[4].value;
}
//Проверка на введенное время
for (i = 0; i < countTask; i++) {
    //Проверка на повторение заданного времени
    for (j = (countTask-1); j > i; j--)
        if (arrTask[i][1] > arrTask[j][0] && arrTask[i][0] < arrTask[j][1]
&& i != j) {
                isTimeRepeat = true;
                var numTimeRepeat = j;
            }
    if (isTimeRepeat) break;
    //Проверка на правильность диапазона времени
    if (arrTask[i][0] >= arrTask[i][1]) {
        isInterval = true;
        var numTimeRepeat = i;
        break;
    }
    //Проверка интервала на обеденное время
    if (arrTask[i][0] < '13:00' && arrTask[i][1] > '12:00') {
        isLunchTime = true;
        numTimeRepeat = i;
        break;
    }
    //Проверка весь ли рабочий день был спланирован
    if (!isNotFullPlanned && arrTask[i][1] != '17:00') {
        if (arrTask[i][1] == '12:00') {

```

```

        for (j = 0; j < countTask; j++)
            if (arrTask[j][0] == '13:00' && i != j) {
                isNotFullPlanned = false;
                break;
            }
        else isNotFullPlanned = true;
    }
    else if (!isNotFullPlanned) {
        for (j = 0; j < countTask; j++)
            if (arrTask[i][1] == arrTask[j][0] && i != j) {
                isNotFullPlanned = false;
                break;
            }
        else isNotFullPlanned = true;
    }
}
//Проверка заполнено ли описание задачи
if (arrTask[i][2] == " && arrTask[i][3] == ") {
    isNopeTask = true;
    numTimeRepeat = i;
    break;
}
//Получение верхнего и нижнего времени планирования
if (arrTask[i][0] < minTime) minTime = arrTask[i][0];
if (arrTask[i][1] > maxTime) maxTime = arrTask[i][1];
}
//Отслеживание ошибок
if (isTimeRepeat) {
    addUserMessage('Повторение времени недопустимо', 'error');
    contentTask[numTimeRepeat].children[0].onclick = function() {
        this.style.border = '1px solid silver';
        this.onclick = null;
    }
    contentTask[numTimeRepeat].children[0].style.border = '1px solid red';
    return;
}
if (isInterval) {
    var error = 'Неверно задан интервал времени';
    addUserMessage(error, 'error');
}

```

Продолжение Приложения Д Листинг программы

```

contentTask[numTimeRepeat].children[1].onclick = function() {
    this.style.border = '1px solid silver';
    this.onclick = null;
}
contentTask[numTimeRepeat].children[1].style.border = '1px solid red';
return;
}
if (isLunchTime) {
    addUserMessage('Была запланирована задача на обеденный перерыв', 'error');
    contentTask[numTimeRepeat].children[0].onclick = function() {
        this.style.border = '1px solid silver';
        this.onclick = null;
    }
    contentTask[numTimeRepeat].children[0].style.border = '1px solid red';
    return;
}
if (isNopeTask) {
    addUserMessage('Одно из полей должно быть заполнено', 'error');
    contentTask[numTimeRepeat].children[2].onclick = function() {
        this.style.border = '1px solid silver';
        this.onclick = null;
    }
    contentTask[numTimeRepeat].children[2].style.border = '1px solid red';
    contentTask[numTimeRepeat].children[3].onclick = function() {
        this.style.border = '1px solid silver';
        this.onclick = null;
    }
    contentTask[numTimeRepeat].children[3].style.border = '1px solid red';
    return;
}
var date = $('#date').val();
var coWorker = $('#co-worker').val();
//Проверка на то, был ли выбран сотрудник
if (coWorker == 0) {
    addUserMessage('Не выбран сотрудник', 'error');
    $('#co-worker')[0].onclick = function() {
        this.style.border = '1px solid silver';
        this.onclick = null;
    }
}

```

Продолжение Приложения Д Листинг программы

```

    $('#co-worker')[0].style.border = '1px solid red';
    return;
}

```

Продолжение Приложения Д Листинг программы

```

//
if (minTime != '08:00' || maxTime != '17:00' || isNotFullPlanned)
    var contin = confirm('Спланированы задачи не на весь день, всеравно
добавить план? Дальнейшее добавление задач будет недоступно');
if (!contin) return;
//Пост запрос
$.ajax({
    url: 'ajax/addTask.php',
    type: 'POST',
    cache: false,
    data: ({date: date, coWorker: coWorker, arrTask: arrTask}),
    dataType: 'html',
    beforeSend: function () {
        $('#btnAddPlan').attr("disabled", "disabled");
    },
    success: function(data) {
        $('#btnAddPlan').removeAttr("disabled");
        if (data == 'success') {
            numberTask = 0;
            $('#co-worker')[0].value = 0;
            $('.timeFuct').remove();
            addTask();
            addUserMessage('Планбылдобавленуспешно',data);
        }
        else {
            addUserMessage(data, 'error');
        }
    }
});

});
</script>

</body>
</html>

```

Автоматическое планирование рабочего дня:

```
function autoPlan(listPlanUser, listPlanAnother, listPlanFile) {
    numberTask = 0;

    var taskDiv = $('#addTask')[0].children;
    for (var i = (taskDiv.length-1); i > 2 ; i--) {
        taskDiv[i].remove();
    }
    task = 'task-' + numberTask;
    if (document.getElementById(task).childNodes[1].value == '17:00') return;
    var task, block, length, sumLen = 1;
    var idUser = $('formPlan')[0][0].value;
    //Автоматическое планирования исполнителем
    if (listPlanUser) {
        //Выбор из списка невыполненного ранее сотрудником
        length = listPlanUser.length;
        for (var i = 0; i < length; i++) {
            if (listPlanUser[i].time_with == '16:00:00' || listPlanUser[i].time_with == '08:00:00')
                continue;

            if (sumLen >= 7) break;
            addTask();
            task = 'task-' + numberTask;
            block = document.getElementById(task);
            if (block.childNodes[0].value == '16:00') break;
            timeTask = listPlanUser[i].plan_time - listPlanUser[i].accounting_hours;
            if (timeTask <= 0) timeTask = 1;
            if (timeTask > 1) {
                var time = block.childNodes[0].value;
                block.childNodes[1].value = sumHours(time, listPlanUser[i].plan_time);
            }
            if (block.childNodes[1].value > '16:00') {
                block.remove();
                numberTask--;
                continue;
            }
            block.childNodes[2].value = listPlanUser[i].title;
            block.childNodes[3].value = listPlanUser[i].description;
            block.childNodes[4].value = timeTask;
            sumLen += timeTask;
        }
    }
}
```

Продолжение Приложения Д Листинг программы

```
//Выборизспискаобщие
var id = '<?=$_SESSION['logged_user'][TASKS_ID]?>';
length = listPlanAnother.length;
```

ПродолжениеПриложенияД Листингпрограммы

```
for (var i=0; i < length; i++) {
    if (id == listPlanAnother[i].id_user || listPlanAnother[i].time_with == '16:00:00' ||
listPlanAnother[i].time_with == '08:00:00') continue;
    if (sumLen >= 7 ) break;
    addTask();
    task = 'task-' + numberTask;
    block = document.getElementById(task);
    if (block.childNodes[0].value == '16:00') break;
    timeTask = listPlanAnother[i].plan_time - listPlanAnother[i].accounting_hours;
    if (timeTask <= 0) timeTask = 1;
    if (timeTask > 1) {
        var time = block.childNodes[0].value;
        block.childNodes[1].value = sumHours(time, listPlanAnother[i].plan_time);
    }
    if (block.childNodes[1].value > '17:00') {
        block.remove();
        numberTask--;
        continue;
    }
    block.childNodes[2].value = listPlanAnother[i].title;
    block.childNodes[3].value = listPlanAnother[i].description;
    block.childNodes[4].value = timeTask;
    sumLen += timeTask;
}
}
//Автоматическое планирование руководителем
else {
    if (idUser == 0) {
        addUserMessage('Не выбран сотрудник', 'error');
        $('#co-worker')[0].onclick = function() {
            this.style.border = '1px solid silver';
            this.onclick = null;
        }
        $('#co-worker')[0].style.border = '1px solid red';
        return;
    }
}
```

```
length = listPlanAnother.length;
```

```
for (var i=0; i < length; i++) {
```

Продолжение Приложения Д Листинг программы

```
    if ((idUser != listPlanAnother[i].id_user && listPlanAnother[i].id_notation != 4) ||  
(listPlanAnother[i].id_notation == 4 && idUser == listPlanAnother[i].id_user) || listPlanAnother[i].time_with ==  
'16:00:00' || listPlanAnother[i].time_with == '08:00:00') continue;
```

```
    if (sumLen >= 7) break;
```

```
    addTask();
```

```
    task = 'task-' + numberTask;
```

```
    block = document.getElementById(task);
```

```
    if (block.childNodes[0].value == '16:00') break;
```

```
    timeTask = listPlanAnother[i].plan_time - listPlanAnother[i].accounting_hours;
```

```
    if (timeTask <= 0) timeTask = 1;
```

```
    if (timeTask > 1) {
```

```
        var time = block.childNodes[0].value;
```

```
        block.childNodes[1].value = sumHours(time, listPlanAnother[i].plan_time);
```

```
    }
```

```
    if (block.childNodes[1].value > '17:00') {
```

```
        block.remove();
```

```
        numberTask--;
```

```
        continue;
```

```
    }
```

```
    block.childNodes[2].value = listPlanAnother[i].title;
```

```
    block.childNodes[3].value = listPlanAnother[i].description;
```

```
    block.childNodes[4].value = timeTask;
```

```
    sumLen += timeTask;
```

```
    }
```

```
}
```

```
//Выбор из списка файла
```

```
length = listPlanFile.length;
```

```
for (var i=0; i < length; i++) {
```

```
    if (sumLen >= 7) break;
```

```
    addTask();
```

```
    task = 'task-' + numberTask;
```

```
    block = document.getElementById(task);
```

```
    if (block.childNodes[0].value == '16:00') break;
```

```
    timeTask = listPlanFile[i].plan_time - listPlanFile[i].accounting_hours;
```

```
    if (timeTask <= 0) timeTask = 1;
```

```
    if (timeTask > 1) {
```

```
        var time = block.childNodes[0].value;
```

```

        block.childNodes[1].value = sumHours(time, listPlanFile[i].plan_time);
    }

```

Продолжение Приложения Д Листинг программы

```

    if (block.childNodes[1].value > '16:00') {
        block.remove();
        numberTask--;
        continue;
    }
    block.childNodes[2].value = listPlanFile[i].title;
    block.childNodes[3].value = listPlanFile[i].description;
    block.childNodes[4].value = timeTask;
    sumLen += timeTask;
}
if (sumLen >= 7) {
    addTask();
    return;
}
}

```

Страница добавление сотрудника:

```

<?php
    require_once "session.php";
    if(!isset($_SESSION['logged_user']) || $_SESSION['logged_user'][USER_ID_GROUP] != 1):
?>
    <script>document.location.href="/" </script>
<? endif;?>
<!DOCTYPE html>
<html lang="ru">
<head>
    <?php
        $title = "Добавление сотрудника | ВКР";
        require_once "block/head.php";
    ?>
</head>
<body>
    <!-- Добавление пользователя (POST запрос) -->
    <?php require_once "block/addUser.php" ?>
    <!--Шапка -->
    <?php require_once "block/header.php" ?>
    <!--Контент -->

```



```
<div class="container-fluid">
```

Продолжение Приложения Д Листинг программы

```
<div class="row">
  <!--Миникартасайта -->
  <div class="map-site-bg">
    <nav aria-label="breadcrumb">
      <ol class="breadcrumb">
        <li class="breadcrumb-item"><a href="/">Главнаястраница</a></li>
          <li class="breadcrumb-item"><a href="/listCo-
worker.php">Списоксотрудников</a></li>
            <li class="breadcrumb-item active" aria-
current="page">Добавлениесотрудника</li>
          </ol>
        </nav>
      </div>
      <div class="reg-user">
        <div class="titleReg">Добавление сотрудника</div>
        </div>
        <form action="" method="post" class="formReg">
          <input type="hidden" name="id_leader" id="id_leader"
value="<?=$_SESSION['logged_user']['USER_ID']?>">
            <label>Фамилия<input type="text" placeholder="Иванов" name="last_name"
id="last_name" required value="<?=$last_name?>"></label>
            <label>Имя<input type="text" placeholder="Иван" name="first_name"
id="first_name" required value="<?=$first_name?>"></label>
            <label>Отчество<input type="text" placeholder="Игоревич"
name="middle_name" id="middle_name" required value="<?=$middle_name?>"></label>
            <label>Email <input type="email" placeholder="example@mail.ru"
name="email" id="email" required value="<?=$email?>"></label>
          <br>
            <label>Пароль<input type="password" placeholder="8fGh34ejK" name="pssw"
id="pssw" required value="<?=$password?>"></label>
          <br>
            <label>Повторитепароль<input type="password" placeholder="8fGh34ejK"
name="r_pssw" id="r_pssw" required></label>
          <?
            if($mssg)
              echo
'<script>$(document).ready(function(){addUserMessage('\.$mssg.', '\error\');});</script>';
          ?>
```

```

        <divclass="btn-center">
            Продолжение Приложения Д
            Листингпрограммы
            <button type="submit" class="btn btn-outline-success" id="regist"
name="regBbtn">Добавить</button>
        </div>
    </form>
</div>
</div>
<!-- Подвал -->
<?php require_once "block/footer.php" ?>
</body>
</html>

```

ФУНКЦИИ:

```

function connectDB () {
    $host = 'localhost';
    $database = 'gnewsport_VKR';
    $user = 'root';
    $password = "";
    global $mysqli;
    $mysqli = new mysqli($host, $user, $password, $database);
    if ($mysqli->connect_error) {
        printf("Connect failed: %s\n", $mysqli->connect_error);
        exit();
    }
    $mysqli->query( "SET CHARSET utf8" );
}

function closeDB () {
    global $mysqli;
    $mysqli->close();
}

function getUser($email,$id_leader) {
    $where = "";
    global $mysqli;
    connectDB();
    if ($email)
        $where = "WHERE `email` LIKE ".$email."";
    if ($id_leader)
        $where = "WHERE ".USER_ID_LEADER."=".$id_leader ORDER BY ".USER_LAST_NAME."
ASC";

```

```
$result = $mysqli->query("SELECT * FROM `users` $where");
```

Продолжение Приложения Д Листинг программы

```
closeDB();
if ($email) {
    if ($result)
        return $result->fetch_assoc();
    else
        return NULL;
}
else {
    return resultToArray($result);
}
}

function getUserByID($id) {
    global $mysqli;
    connectDB();
    $result = $mysqli->query("SELECT * FROM ".TABLE_USER." WHERE ".USER_ID."=$id");
    closeDB();
    if ($result)
        return $result->fetch_assoc();
    else
        return "";
}

//Функция возвращает активный/неактивный пользователь
function getListUserStatus() {
    global $mysqli;
    connectDB();
    $result = $mysqli->query("SELECT * FROM ".TABLE_USER." WHERE 1 ORDER BY
".USER_CONFIRMATION." ASC");
    closeDB();
    return resultToArray($result);
}

function getListGroup() {
    global $mysqli;
    connectDB();
    $result = $mysqli->query("SELECT * FROM ".TABLE_USER_GROUP." WHERE 1");
    closeDB();
    return resultToArray($result);
}

function getNameGroup($id) {
```

```
global $mysqli;
```

Продолжение Приложения Д Листинг программы

```
connectDB();
$result = $mysqli->query("SELECT * FROM ".TABLE_USER_GROUP." WHERE
".USER_GROUP_ID."=$id");
closeDB();
if ($result)
    return $result->fetch_assoc()[USER_GROUP_TITLE];
else
    return "";
}
function getIdGroupUser($nameGroup) {
    global $mysqli;
    connectDB();
    $result = $mysqli->query("SELECT * FROM ".TABLE_USER_GROUP." WHERE
".USER_GROUP_TITLE." LIKE '$nameGroup'");
    closeDB();
    if ($result)
        return $result->fetch_assoc()[USER_GROUP_ID];
    else
        return null;
}
//Функция возвращает ФИО пользователя по id
function getFIObyID($id) {
    global $mysqli;
    connectDB();
    $where = "WHERE `id`=$id";
    $result = $mysqli->query("SELECT * FROM `users` $where");
    closeDB();
    return getFIO($result->fetch_assoc());
}
//Функция добавления нового пользователя
function addUser($fName, $mName, $lName, $email, $psw, $idLeader, $idGroup) {
    global $mysqli;
    connectDB();
    $result = $mysqli->query("INSERT INTO ".TABLE_USER."(".USER_FIRST_NAME.",
".USER_LAST_NAME.", ".USER_MIDDLE_NAME.", ".USER_EMAIL.", ".USER_PASSWORD.",
".USER_ID_LEADER.", ".USER_ID_GROUP.") VALUES ('$fName', '$lName', '$mName', '$email', '$psw',
$idLeader, $idGroup)");
    closeDB();
}
```

```
if ($result)
```

Продолжение Приложения Д Листинг программы

```
    return true;
```

```
else
```

```
    return false;
```

```
}
```

```
//Изменениеданныхвтаблице
```

```
function changeDataAny($nameTable, $colArr, $valArr, $id, $colWhere) {
```

```
    if (count($valArr) != count($colArr)) return false;
```

```
    $set = "";
```

```
    if (is_array($colArr)) {
```

```
        for ($i=0; $i<count($colArr); $i++) {
```

```
            if ($i == count($colArr)-1) $set.= $colArr[$i]."=". $valArr[$i]."";
```

```
            else $set .= $colArr[$i]."=". $valArr[$i].", ";
```

```
        }
```

```
    }
```

```
    else $set = $colArr."=". $valArr."";
```

```
    global $mysqli;
```

```
    connectDB();
```

```
    $result = $mysqli->query("UPDATE $nameTable SET $set WHERE $colWhere=$id");
```

```
    closeDB();
```

```
    if ($result)
```

```
        return true;
```

```
    else
```

```
        return false;
```

```
}
```

```
function getListLeader() {
```

```
    global $mysqli;
```

```
    connectDB();
```

```
    $result = $mysqli->query("SELECT * FROM ".TABLE_USER." WHERE id_group=1");
```

```
    closeDB();
```

```
    return resultToArray($result);
```

```
}
```

```
//Функция возвращает ФИО пользователя (Иванов И.И.)
```

```
function getFIO($user) {
```

```
    if ($user == null) return "";
```

```
    $name = preg_split('//u', $user['first_name'], null, PREG_SPLIT_NO_EMPTY);
```

```
    $middle_name = preg_split('//u', $user['middle_name'], null, PREG_SPLIT_NO_EMPTY);
```

```
    return $user['last_name']." ".$name[0].".".$middle_name[0].".";
```

```
}
```

Продолжение Приложения Д Листинг программы

```
//Функциявозвращаетдатыпланирования
function getDatePlan($isGroup, $where, $isAccepted) {
    global $mysqli;
    $group = "";
    if ($isGroup)
        $group = 'GROUP BY '.DATE_PLAN_DATE;
    if ($isAccepted && $where != "")
        $where.= ' AND '.DATE_PLAN_ACCEPTED.='=1';
    else if($isAccepted) $where = 'WHERE '.DATE_PLAN_ACCEPTED.='=1';
    else if ($where != "" && !$isAccepted) $where.= ' AND '.DATE_PLAN_ACCEPTED.='=0';
    else if (!$isAccepted && $where == "") $where = 'WHERE '.DATE_PLAN_ACCEPTED.='=0';
    connectDB();
    $result = $mysqli->query("SELECT * FROM ".TABLE_DATE_PLAN." $where $group ORDER BY
    ".DATE_PLAN_DATE." DESC");
    closeDB();
    if ($result)
        return resultToArray($result);
    else
        return null;
}

//Функциявозвращаетдатувида 25.03.2018
function formatDate($date) {
    if ($date)
        return date('d.m.Y',strtotime($date));
    else
        return '-';
}

//Функция возвращает время вида hh:mm (вместо hh:mm:ss)
function getFormatTime($time) {
    if ($time < 10)
        return substr($time, 1, 4);
    else
        return substr($time, 0, 5);
}

//Функция запросов на добавления записи в таблицу
function getNewEntry($tableName, $columns, $values) {
    if (count($columns) != count($values))
        return false;
    global $mysqli;
```

Продолжение Приложения Д Листинг программы

```
connectDB();
$col = "";
$val = "";
for ($i=0; $i < count($columns); $i++) {
    if ($i == count($columns) - 1) {
        $col .= $columns[$i].',';
        $val .= $values[$i].',';
    }
    else if ($i == 0) {
        $col .= '('. $columns[$i].',';
        $val .= '('. $values[$i].',';
    }
    else {
        $col .= $columns[$i].',';
        $val .= $values[$i].',';
    }
}
$result = $mysqli->query("INSERT INTO $tableName $col VALUES $val");
$idNewEntry = $mysqli->insert_id;
closeDB();
return $idNewEntry;
}

function getIdDatePlan($date,$id) {
    global $mysqli;
    connectDB();
    $result = $mysqli->query("SELECT * FROM ".TABLE_DATE_PLAN." WHERE
\".DATE_PLAN_ACCEPTED.\"=1 AND \".DATE_PLAN_DATE.\"='\"$date\"' AND
\".DATE_PLAN_ID_USER.\"='\"$id\"'");
    closeDB();
    if ($result)
        return $result->fetch_assoc();
    else
        return NULL;
}

//Функция возвращает список планирования
function getPlan($idDate, $where, $typeTime, $byGroup) {
    global $mysqli;
    if ($byGroup==TASK_TITLE)
```

Продолжение Приложения Д Листинг программы

```
$group = 'GROUPBY'.TASK_TITLE;
else if ($byGroup==TASK_DESCRIPTION)
    $group = 'GROUP BY '.TASK_DESCRIPTION;
else if ($byGroup==TASK_ADMISSION)
    $group = 'GROUP BY '.TASK_ADMISSION;
connectDB();
if ($idDate) $idDateWhere = TASKS_ID_DATEPLAN."=$idDate";
if ($typeTime != null) $typeTimeWhere = "AND ".TASK_TYPE_TIME."=$typeTime";
$result = $mysqli->query("SELECT * FROM ".TABLE_TASKS." WHERE $idDateWhere
$typeTimeWhere $where $group ORDER BY ".TASK_WITH." ASC");
closeDB();
if ($result)
    return resultToArray($result);
else
    return "Поданномузапросунетданных";
}
function getSelectPlan($numActualTime,$plan) {
    for ($i = 0; $i < count($plan); $i++) {
        if ($numActualTime == $plan[$i][TASK_ID_ACTUALTIME])
            return $plan[$i];
    }
    return null;
}
//Функция возвращает список примечаний
function getListNotation() {
    global $mysqli;
    connectDB();
    $result = $mysqli->query("SELECT * FROM ".TABLE_NOTATION_TASK." WHERE 1");
    closeDB();
    if ($result)
        return resultToArray($result);
    else
        return "Данныебылинайдены";
}
//Функция возвращает наименование примечания
function getNameNotation($id) {
    global $mysqli;
    connectDB();
```


Продолжение Приложения Д Листинг программы

```
$result = $mysqli->query("SELECT * FROM ".TABLE_NOTATION_TASK." WHERE  
".NOTATION_TASK_ID."=$id");  
closeDB();  
if ($result)  
    return $result->fetch_assoc()[NOTATION_TASK_TITLE];  
else  
    return "";  
}  
  
// Функция изменение записи данных  
function changeDataTask($nameCol, $val, $id) {  
    global $mysqli;  
    connectDB();  
    $result = $mysqli->query("UPDATE ".TABLE_TASKS." SET $nameCol='$val' WHERE  
".TASKS_ID."=$id");  
    closeDB();  
    if ($result)  
        return true;  
    else  
        return false;  
}  
  
//  
function addDataTask($col, $val, $id, $typeTime) {  
    global $mysqli;  
    connectDB();  
    $result = $mysqli->query("INSERT INTO ".TABLE_TASKS."("TASKS_ID_DATEPLAN.",  
$col,"TASK_TYPE_TIME.") VALUES ($id, '$val',$typeTime)");  
    closeDB();  
    if ($result)  
        return true;  
    else  
        return false;  
}  
  
function getNotDonePlan($idUser, $idGroup) {  
    if ($idUser && $idGroup!=1)  
        $where = " AND t1."DATE_PLAN_ID_USER."=$idUser AND t1."TASK_NOTATION."!=1  
AND t1."TASK_NOTATION."!=4";  
    else if (!$idUser && $idGroup!=1)  
        $where = " AND t1."TASK_NOTATION."=4";  
    elseif ($idGroup == 1)
```

Продолжение Приложения Д Листинг программы

```
$where = " AND t1.".TASK_NOTATION."!=1";
global $mysqli;
connectDB();
$result = $mysqli->query("SELECT * FROM (SELECT * FROM (SELECT
".TABLE_TASKS.".*,".TABLE_DATE_PLAN.".".DATE_PLAN_DATE.",".TABLE_DATE_PLAN.".".DATE_P
LAN_ID_USER." FROM ".TABLE_DATE_PLAN.",".TABLE_TASKS." WHERE
".TABLE_DATE_PLAN.".".DATE_PLAN_ID."=".TABLE_TASKS.".".TASKS_ID_DATEPLAN." AND
".TABLE_TASKS.".".TASK_NOTATION."!=5 AND
".TABLE_DATE_PLAN.".".DATE_PLAN_ACCEPTED."=1 ORDER BY
".TABLE_DATE_PLAN.".".DATE_PLAN_DATE." DESC) t WHERE 1 GROUP BY t.".TASK_TITLE.",
t.".TASK_DESCRIPTION.") t1 WHERE t1.".TASK_TYPE_TIME."=1 $where AND
t1.".DATE_PLAN_DATE."!=".date('Y-m-d')."");
closeDB();
if ($result)
    return resultToArray($result);
else
    return null;
}
function getNotation($id) {
    $where = '1';
    if ($id)
        $where = NOTATION_TASK_ID."=$id";
    global $mysqli;
    connectDB();
    $result = $mysqli->query("SELECT * FROM ".TABLE_NOTATION_TASK." WHERE $where");
    closeDB();
    if ($id)
        return $result->fetch_assoc()[NOTATION_TASK_TITLE];
    else {
        if ($result)
            return resultToArray($result);
        else
            return null;
    }
}
//Функциявозвращениефактическоговремени
function getTimeToString($timeWith, $timeBefore, $typeTime) {
    if ($typeTime == 2) return 'внеплан.';
    else if ($typeTime == 3) return 'офиц. ремонт';
}
```

Продолжение Приложения Д Листинг программы

```
else if ($typeTime == 1) return getFormatTime($timeWith).'.'.getFormatTime($timeBefore);
}
function resultToArray ($result) {
    $array = array();
    while (($row = $result->fetch_assoc()) != false)
        $array[] = $row;
    return $array;
}
```


Этап планирования

Заполняется исполнителем/ руководителем (План)					
Дата	День недели	ФИО	Время, факт	Задача	Описание
26.06.2018	Вт		08.00- 09.00	Планерка, анализ дефектов, оценка эффективности проведенной работы	
			09.00- 10.00		
			10.00- 11.00		
			11.00- 12.00		
			13.00- 14.00		
			14.00- 15.00		
			15.00- 16.00		
			16.00- 17.00	Заполнение журналов, отметки об устранении дефектов, планирование	

Этап ведения отчетности

Заполняется исполнителем (Факт)		Заполняется руководителем
Результат	Обоснование	Примечание
Выполнено		Подтверждено
Не выполнено		Не подтверждено
		Перенесено
		Снято

Выставление часов учета и допуска

Заполняется исполнителем (План+Факт)		
Допуск	Учет рем.	Часы учета
	да	1
	нет	

Пример заполненной таблицы

Заполняется исполнителем/ руководителем (План)						Заполняется исполнителем (Факт)		Заполняется руководителем	Заполняется исполнителем (План+Факт)			Заполняется оперативником					
Дата	День недели	ФИО	Время, факт	Задача	Описание	Результат	Обоснование	Примечание	Допуск	Учет рем.	Часы учета	Факт час. с доп.	Наим. допуска				
20.06.2018	Ср	Иванов С. В.	08.00- 09.00	Планерка, анализ дефектов, оценка эффективности проведенной работы		Выполнено		Подтверждено			1						
			09.00- 10.00	Постановка задачи и запрос данных для выбегов температур КЛЦ		Выполнено		Подтверждено	ИСВ	да	2	6,5					
			10.00- 11.00														
			11.00- 12.00														
			13.00- 14.00											Установка ргохтов на новый сервер	Не выполнено	программный рейд не поддерживается/ к риво встает	Снято
			14.00- 15.00											Запрос учетных данных для администрирования сетевого экрана	Выполнено		Подтверждено
			15.00- 16.00	Настройка СРС новый сервер РЦДУ	Выполнено		Подтверждено	да	2								
			16.00- 17.00	Заполнение журналов, отметки об устранении дефектов, планирование		Выполнено		Подтверждено			1						
			внеплан. офиц.рем.														
			*	Выполнение ранее перенесенных(не подтвержденных) задач	1) Дефектация и сортировка модулей находящихся в ремонте (ЖДиР БСБ 2017.12.17; ВСА 2017.12.20): провести учет изъятых предохранителей в шкафах и нехватки их из комплекта; 2) Организация учета превышения температур металлов	Не выполнено		Перенесено									
			учет часов								8	6,5					

Заполнение оперативным персоналом

Заполняется оперативником	
Факт час. с доп.	Наим. допуска
1	

Атрибуты сущностей СУБД

Задачи
Код
Код даты планирования
Время начала
Время окончания
Задача
Описание
Результат
Обоснование
Код примечания
Допуск
Учет ремонта
Часы учета
Плановое время
Тип времени
Факт. часы с допуском
Наименование допуска

Группа пользователей
Код
Название

Примечание
Код
Заголовок

Дата планирования
Код
Дата
Код пользователя
Подтверждение

Пользователи
Код
Имя
Фамилия
Отчество
Код группы
Почта
Пароль
Код руководителя
Администратор
Подтверждение

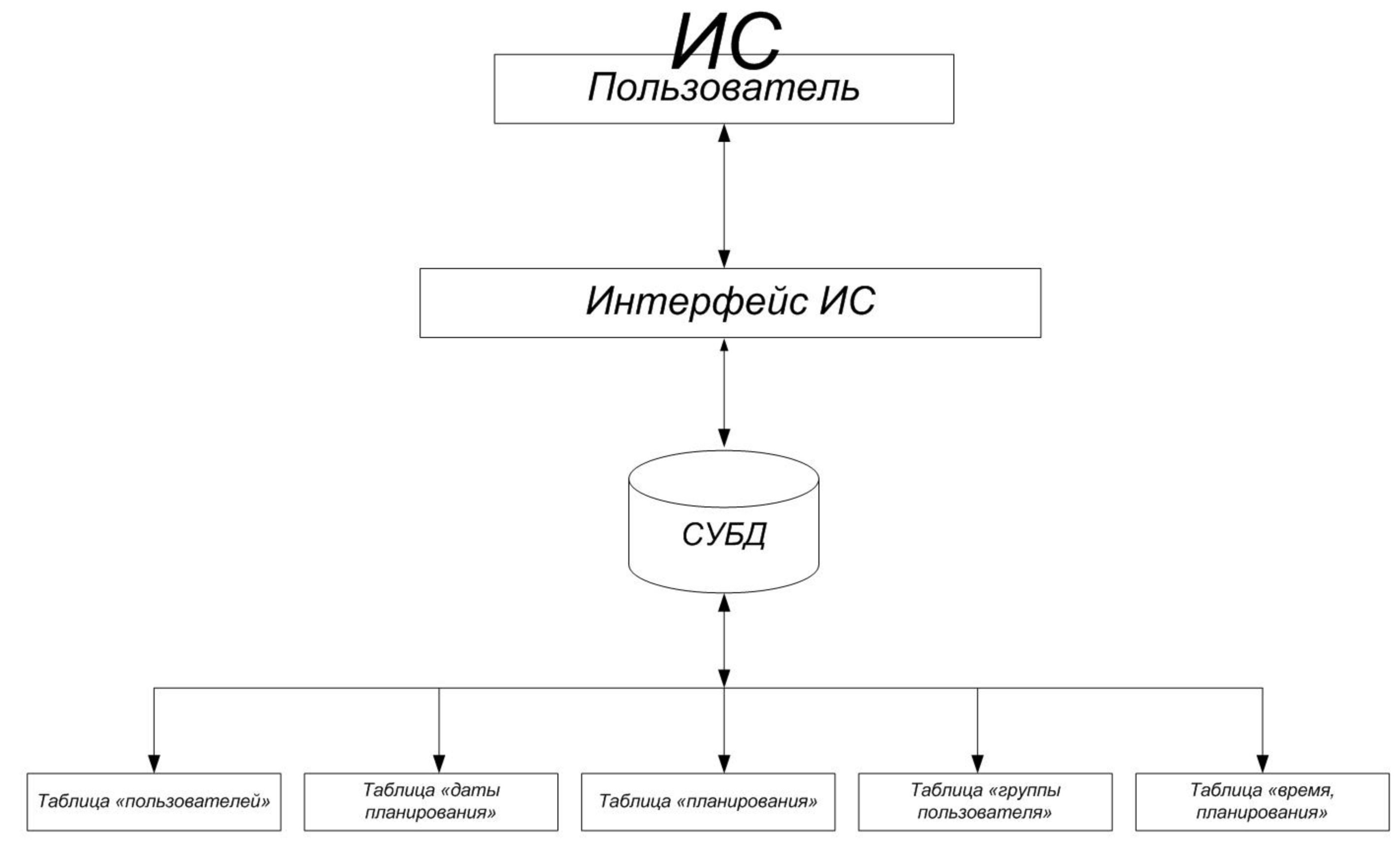
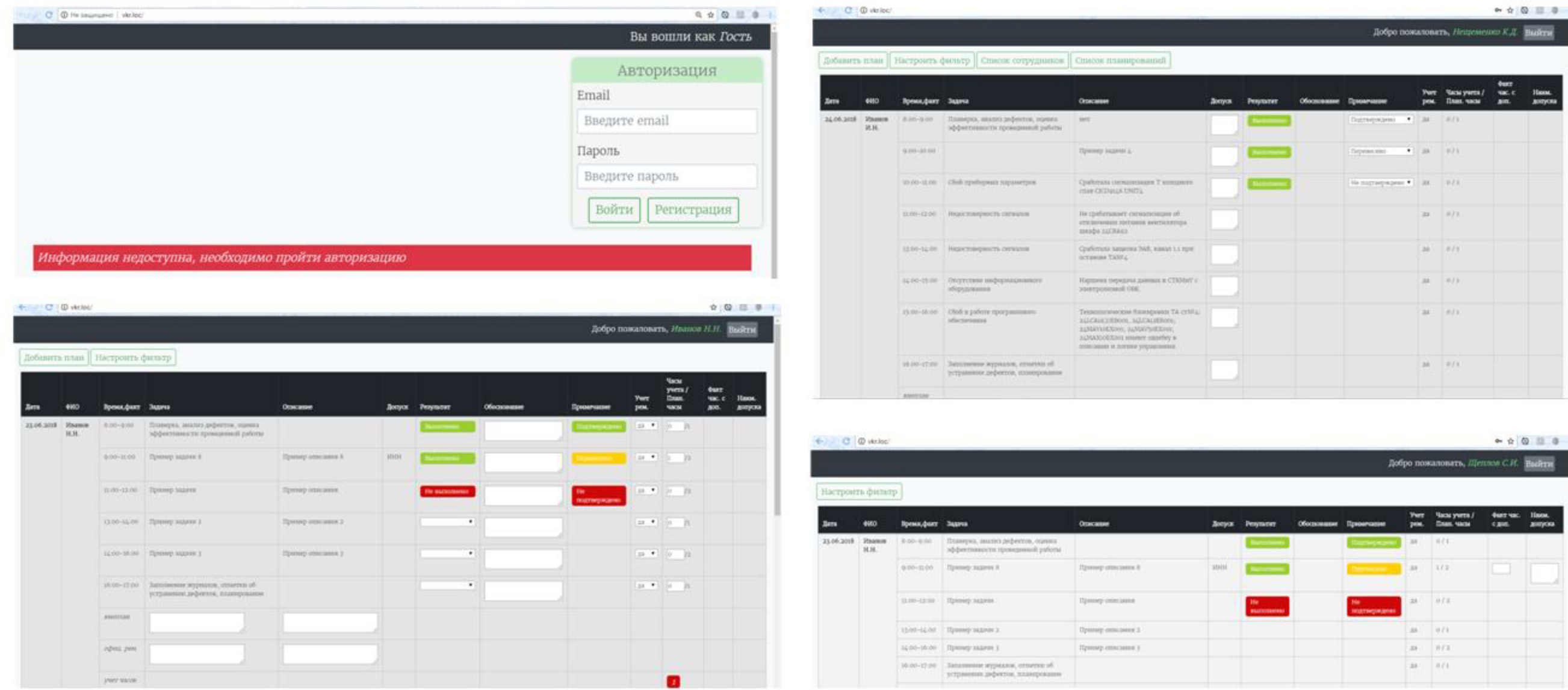


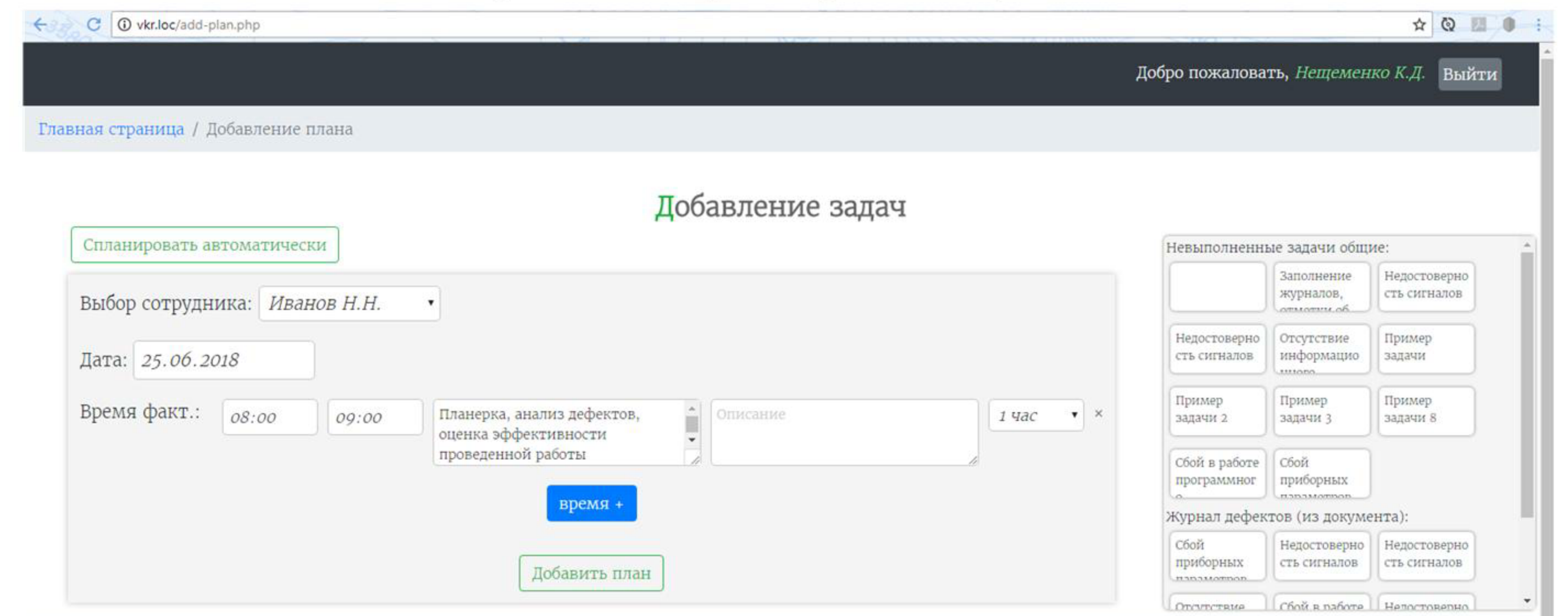
Схема данных СУБД



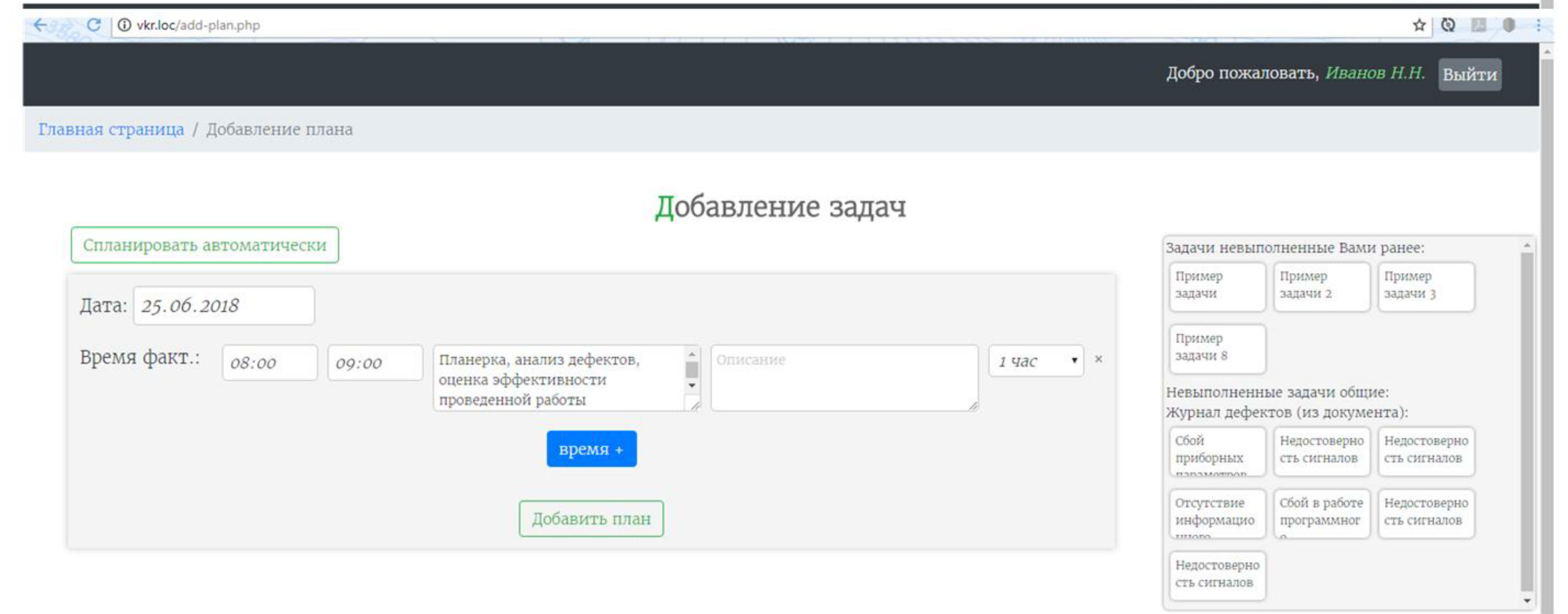
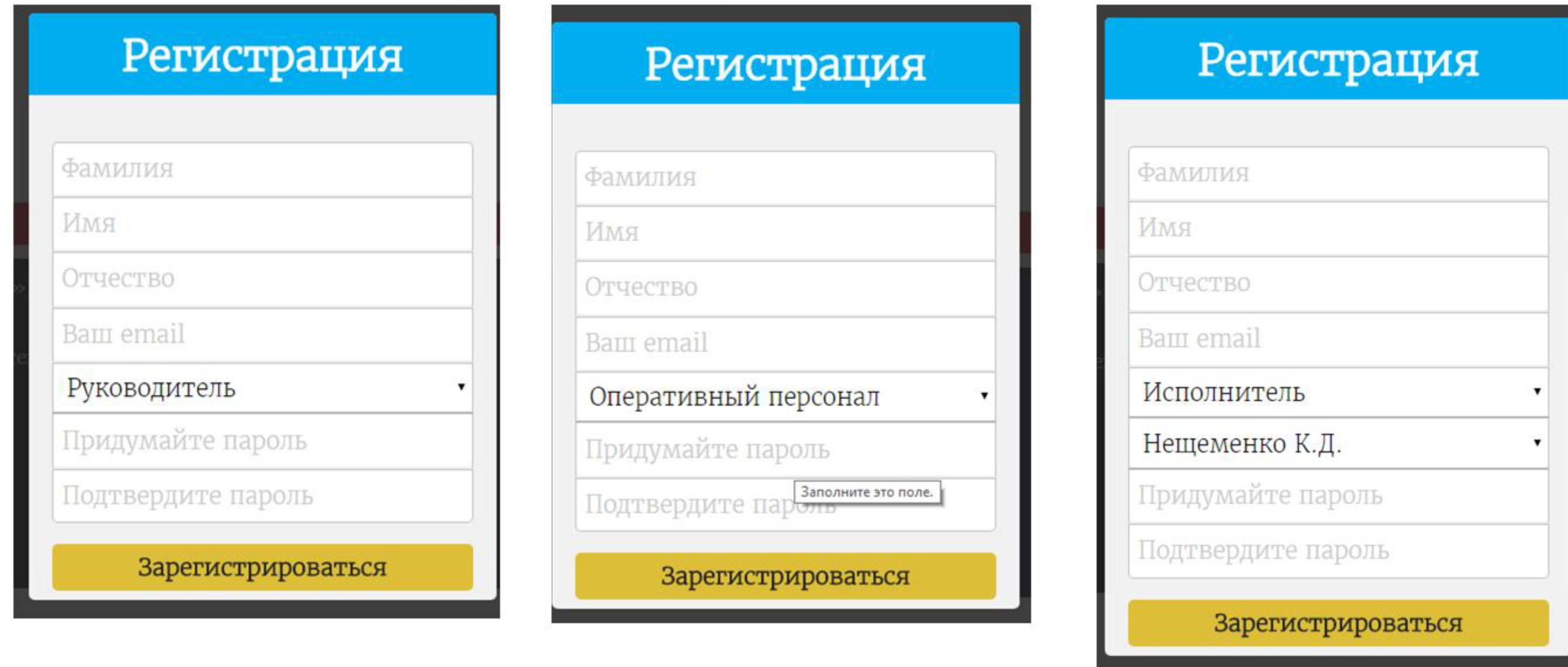
Общий вид главной страницы



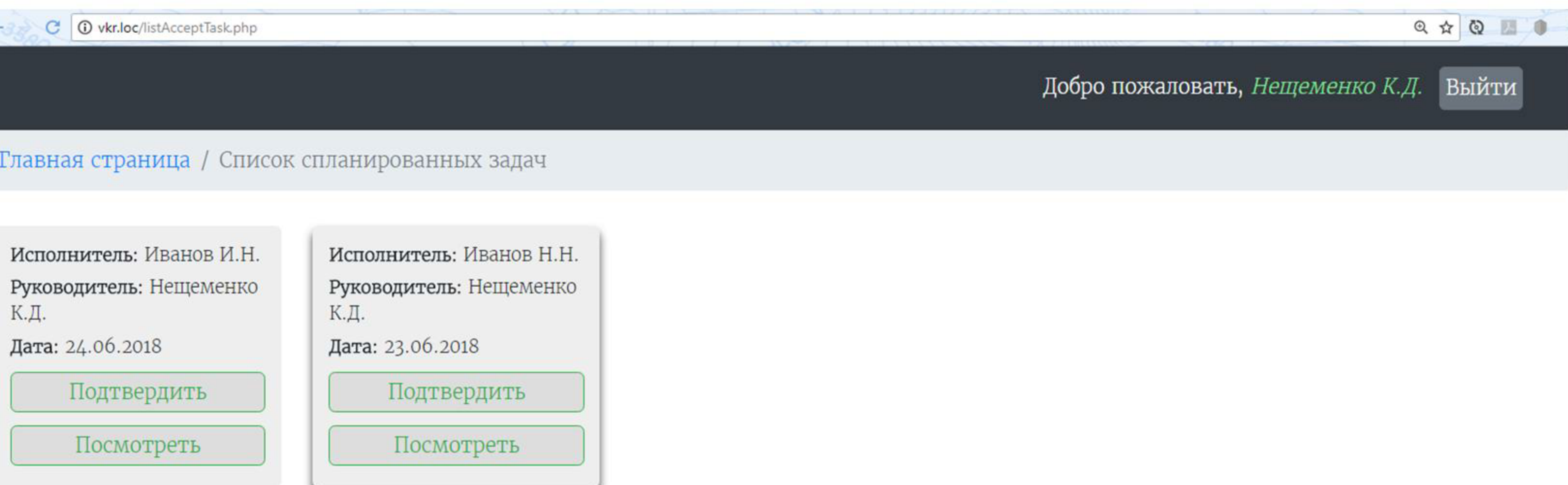
Общий вид страницы планирования



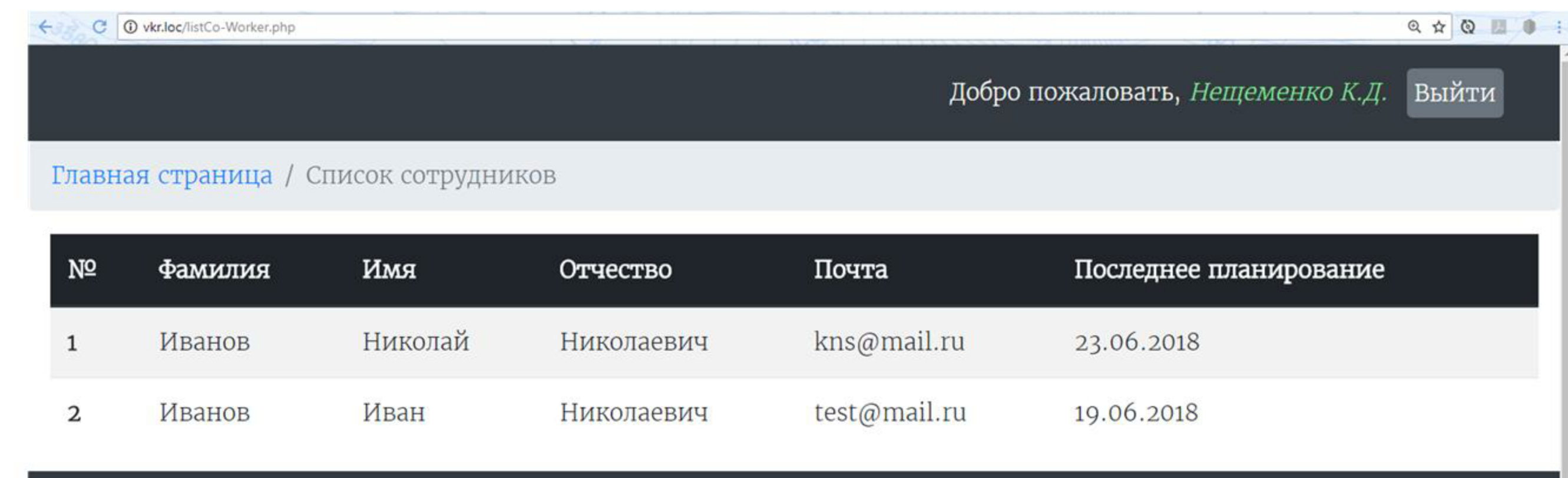
Форма регистрации пользователя



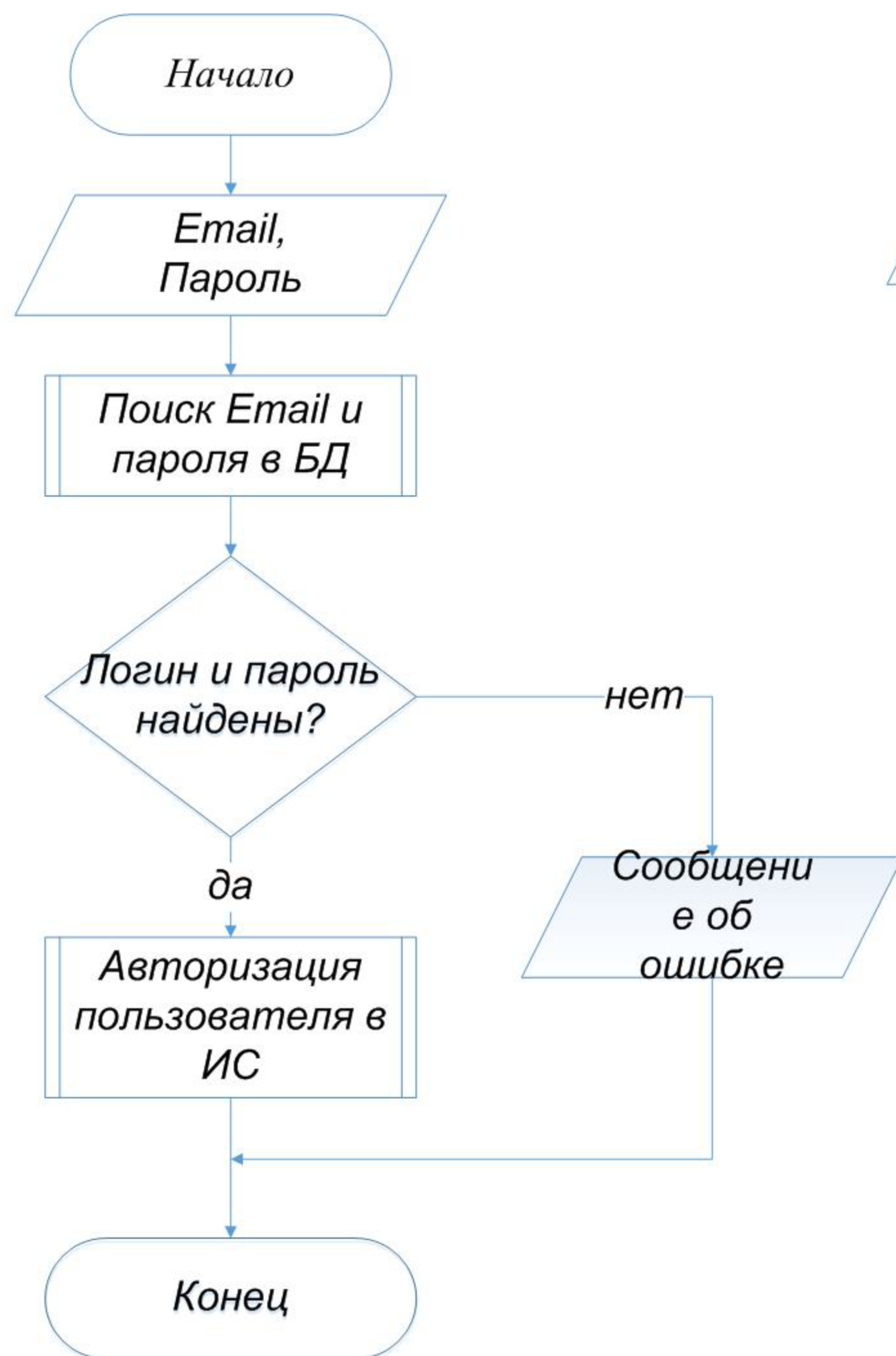
Общий вид страницы подтверждения спланированного дня исполнителем



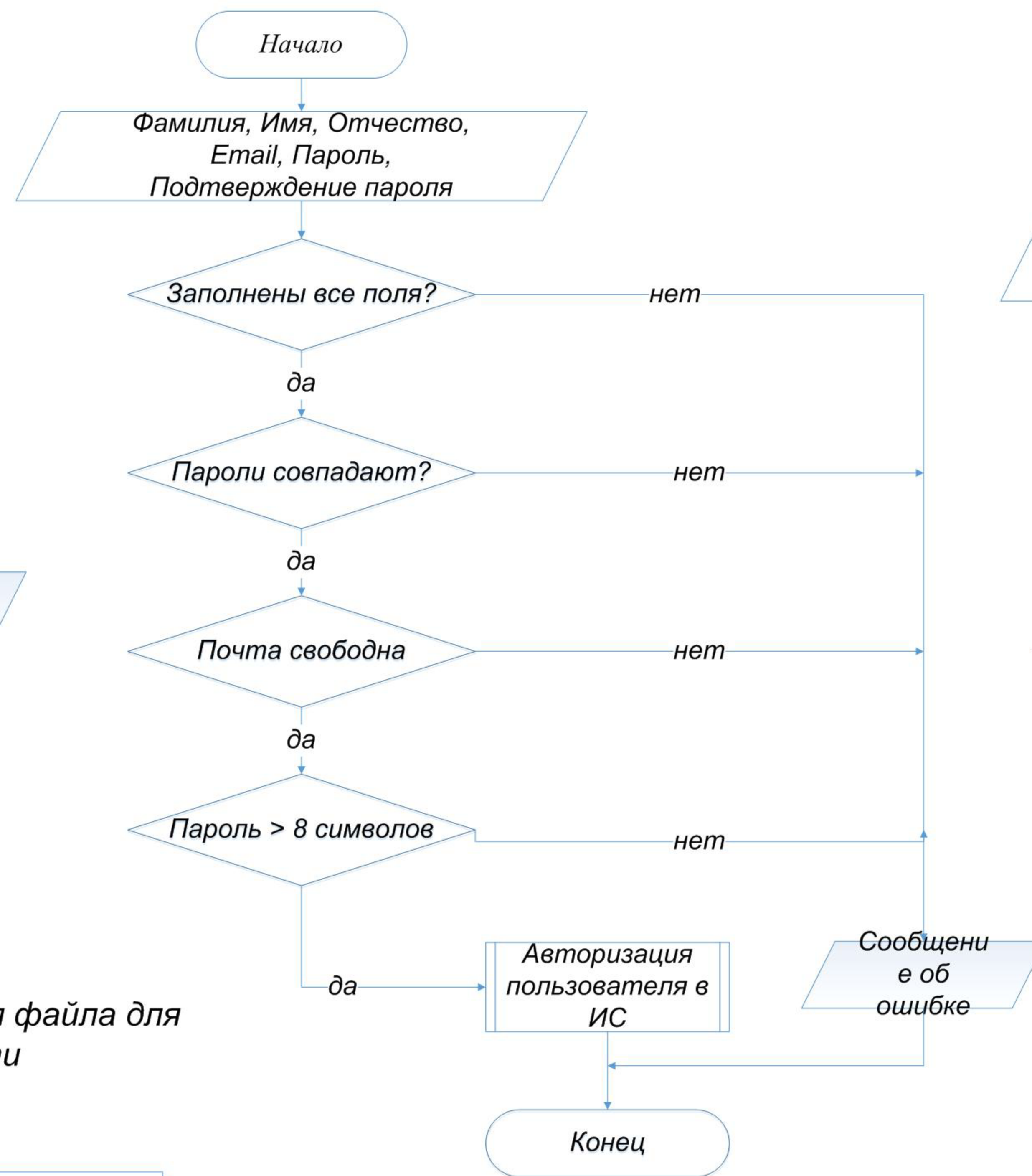
Общий вид страницы списка сотрудников



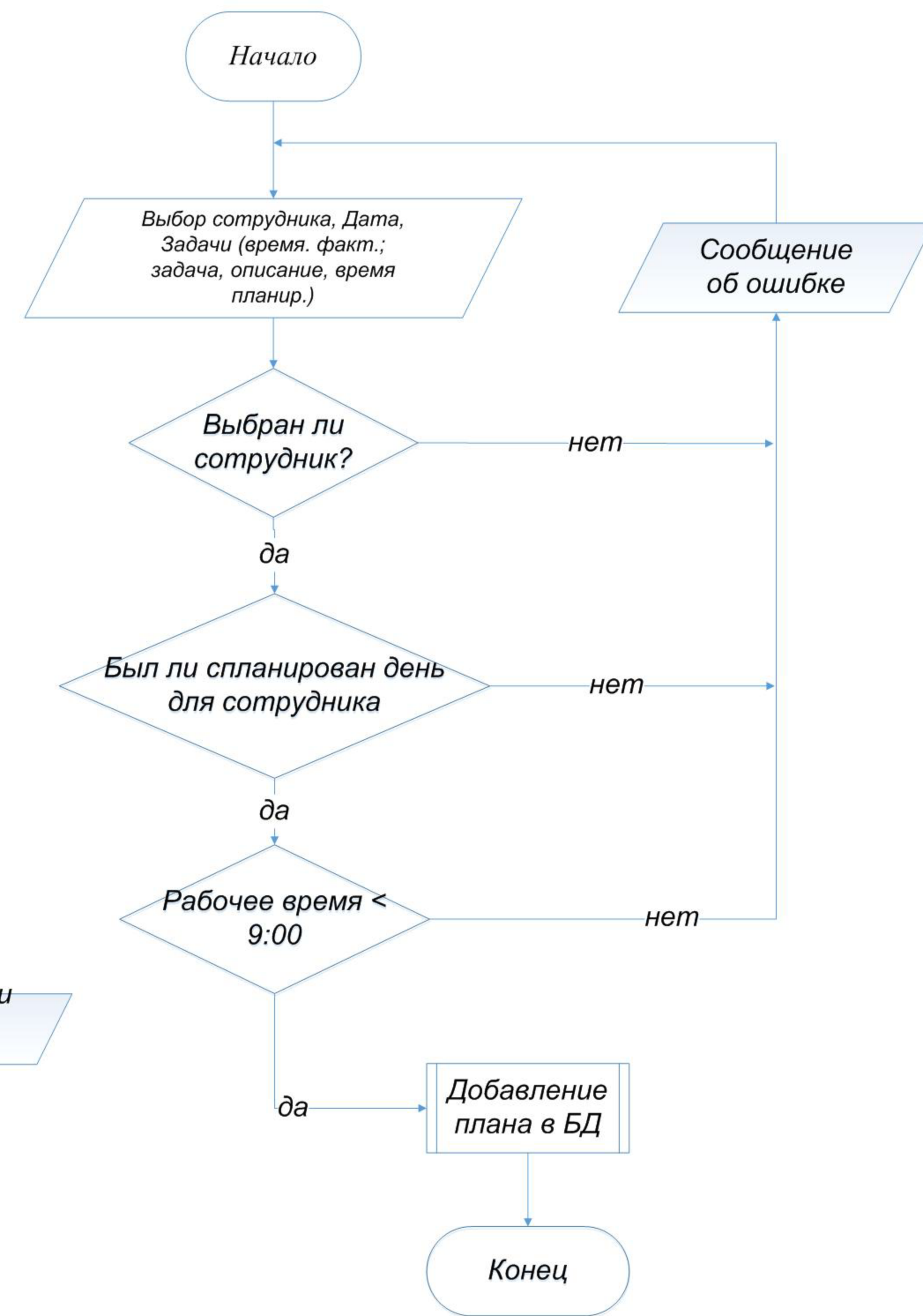
Алгоритм авторизации в системе



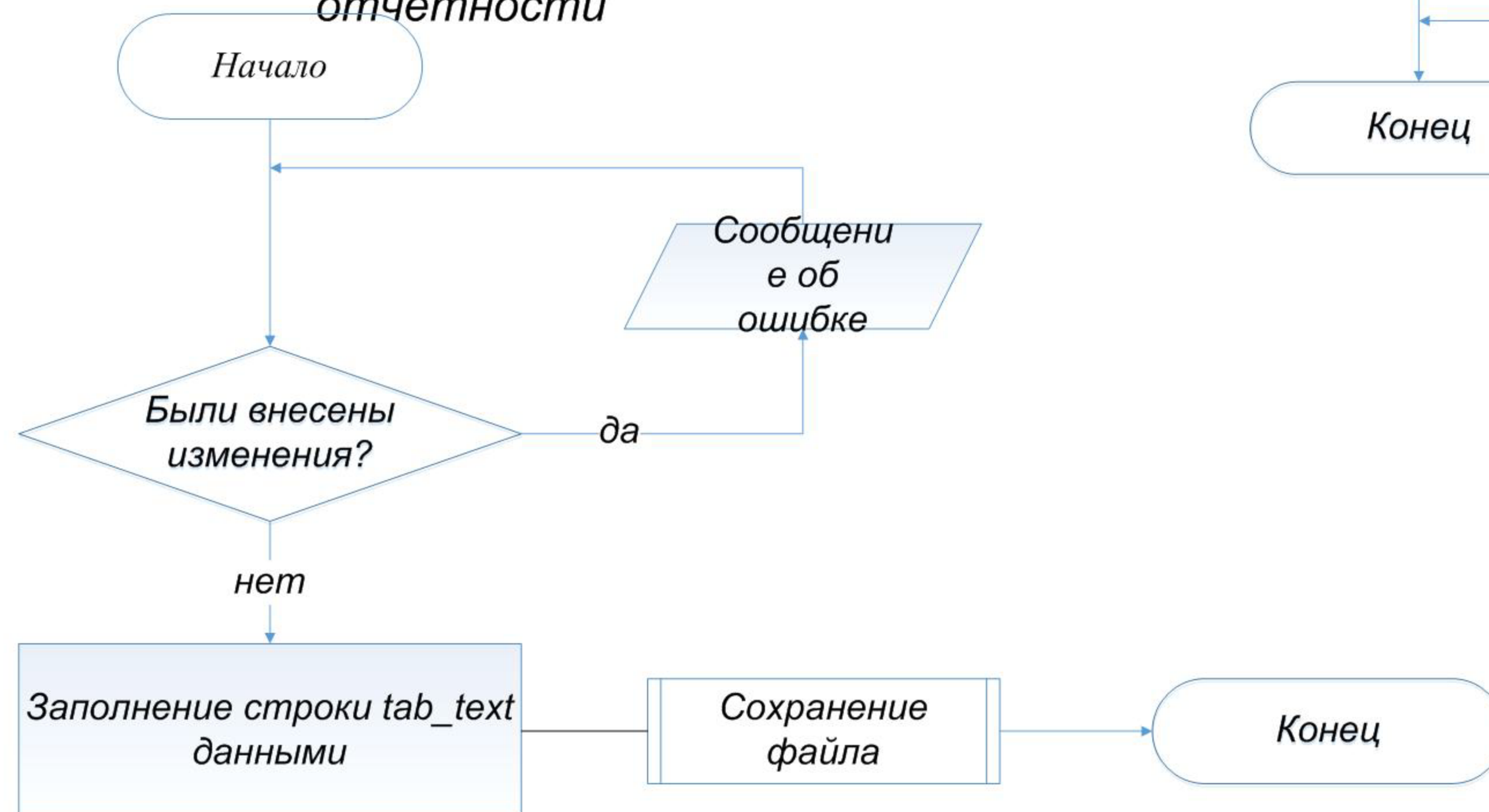
Алгоритм добавление сотрудника



Алгоритм планирования рабочего дня



Алгоритм сохранения файла для отчетности



				ВКР.144002.150304.СХ			
Изм.	Лист	№ докум.	Подпись	Дата	Лит.	Масса	Масштаб
Разраб.	у	Афанасов Л.С.					б
Проверил		Телешко Д.А.					
Т.контр.		Телешко Д.А.			Лист 4		Листов 6
Н.контр.		Скрипко О.В.			Создание автоматизированной системы планирования и учета работ по АСУ ТП на Благовецкой ТЭЦ (комплексная выпускная квалификационная работа)		АМГУ гр.441 об
Уте.		Скрипко О.В.					

Процесс регистрации

Процесс редактирования данных пользователей

Процесс авторизации

Процесс подтверждения регистрации

Регистрация

Афанасов

Леонид

Сергеевич

als@mail.ru

Руководитель

.....

.....

Зарегистрироваться

- ✔ **Успешно!**
 Вы успешно зарегистрировались в системе, ожидайте подтверждение регистрации
- ! **Ошибка!**
 Введите фамилию
- ! **Ошибка!**
 Email введен неверно
- ! **Ошибка!**
 Пароли не совпадают
- ! **Ошибка!**
 Данная почта уже используется
- ! **Ошибка!**
 Пароль слишком короткий. Минимум 9 символов.

kns@mail.ru Иванов Николай Николаевич Исполнитель Иещеменко К.Д. ✕ подтвержден

kns@mail.ru Иванов Николай Николаевич Руководитель ✕ подтвержден

✔ **Успешно!**
 Изменения внесены успешно

Авторизация

Email

admin@mail.ru

Пароль

.....

Войти
Регистрация

- ! **Ошибка!**
 Пользователь с данным email не найден.
- ! **Ошибка!**
 Пароль введен неверно.

✔ **Успешно!**
 Регистрация пользователя подтверждена

✔ **Успешно!**
 Регистрация пользователя была отменена

Подтвердите действие на странице vkr.loc

Подтвердить регистрацию пользователя Афанасов Л. С.

ОК
Отмена

Подтвердите действие на странице vkr.loc

Вы уверены что хотите отменить регистрацию пользователя Афанасов Л. С.

ОК
Отмена

email	фамилия	Имя	Отчество	Права доступа	Руководитель (ФИО)	Статус
als@mail.ru new	Афанасов	Леонид	Сергеевич	Руководитель		+ ✕
admin@mail.ru	<input type="text" value="Афанасов"/>	<input type="text" value="Леонид"/>	<input type="text" value="Сергеевич"/>	Администратор		подтвержден

				ВКР.144002.150304.		
				Сх		
Изм.	Лист	№ докум.	Подпись	Дата		
		Афанасов Л.С.				
Принцип работы информационной системы						
				Лист 5	Листов 6	
Создание автоматизированной системы планирования и учета работ по АСУ ТП на Благовецкой ТЭЦ (комплексная выпускная квалификационная работа)				АМГУ гр.441 об		

Процесс планирования

Спланировать автоматически

Дата: 25.06.2018

Время факт.: 08:00 09:00

08:00	09:00	Планерка, анализ дефектов, оценка эффективности проведенной работы	Описание	1 час
10:00	12:00	Сбой приборных параметров	Сработала сигнализация Т холодного спая CJCD9148.UNIT4	2 часа
11:00	12:00	Недоверность сигналов	Не срабатывает сигнализация об отключении питания вентилятора шкафа 24CRA92	1 час
13:00	15:00	Недоверность сигналов	Сработала защелка ЭАБ, канал 1.1 при останове TAN ^{№4}	2 часа
15:00	16:00	Сбой в работе программного обеспечения	Технологические блокировки ТА ст ^{№4} : 24LCA11(2)EB001, 24LCA12EB001; 24MAV10EX001;	1 час
16:00	17:00	Заполнение журналов, отметки об устранении дефектов, планирование	Описание	1 час

время +

Добавить план

Невыполненные задачи общие:

Заполнение журналов, отметки об	Пример задачи
Пример задачи 2	Пример задачи 3
Пример задачи 8	

Журнал дефектов (из документа):

Сбой приборных параметров	Недоверность сигналов	Недоверность сигналов
Отсутствие информации	Сбой в работе программного обеспечения	Недоверность сигналов
Недоверность сигналов		

Успешно!
План был успешно добавлен, ожидайте ответа от руководителя

- Ошибка!** Не выбран сотрудник
- Ошибка!** Одно из полей должно быть заполнено
- Ошибка!** Была спланирована задача на обеденный перерыв
- Ошибка!** Повторение времени недопустимо
- Ошибка!** Неверно задан интервал времени

Процесс подтверждения спланированного дня

Исполнитель: Иванов И.Н.

Дата: 25.06.2018

Время факт.: 08:00 09:00

08:00	09:00	Планерка, анализ дефектов, оценка эффективности проведенной работы	допуск	1 час
09:00	11:00	Сбой приборных параметров	допуск	2 часа
11:00	12:00	Недоверность сигналов	допуск	1 час
13:00	15:00	Недоверность сигналов	допуск	2 часа
15:00	16:00	Сбой в работе программного обеспечения	допуск	1 час
16:00	17:00	Заполнение журналов, отметки об устранении дефектов, планирование	допуск	1 час

Добавить план

Исполнитель: Иванов И.Н.
Руководитель: Нещепенко К.Д.
Дата: 25.06.2018

Подтвердить

Посмотреть

Исполнитель: Иванов И.Н.
Руководитель: Нещепенко К.Д.
Дата: 25.06.2018

Подтвердить

Посмотреть

Исполнитель: Иванов И.Н.
Руководитель: Нещепенко К.Д.
Дата: 24.06.2018

Подтвердить

Посмотреть

Исполнитель: Иванов И.Н.
Руководитель: Нещепенко К.Д.
Дата: 23.06.2018

Подтвердить

Посмотреть

Успешно!
План был успешно подтвержден

				ВКР.144002.150304.		
				Сх		
Изм.	Лист	№ докум.	Подпись	Дата	Лит.	Масса
Разраб.	Афанасов Л.С.				у	
Проверил	Тепличко Д.А.					
Т.контр					Лист 6	Листов 6
Н.контр	Скрилко О.В.	Принцип работы информационной системы				
Уте.	Скрилко О.В.	Создание автоматизированной системы планирования и учета работ по АСУ ТП на Благовецкой ТЭЦ (комплексная выпускная квалификационная работа)				АМГУ гр.441 об