Министерство науки и высшего образования Российской Федерации

Федеральное государственное бюджетное образовательное учреждение   
высшего образования

«Иркутский национальный исследовательский технический

университет»

**Институт информационных технологий и анализа данных**

**О Т Ч Ё Т**

|  |  |  |
| --- | --- | --- |
| о прохождении | | Производственной |
|  | | (вид практики: учебная/производственная) |
|  | | |
| (тип практики: технологическая/научно-исследовательская работа/преддипломная и др.) | | |
| практики | | |
| на | ФИЛИАЛ "ГУСИНООЗЕРСКАЯ ГРЭС" АО "ИНТЕР РАО-ЭЛЕКТРОГЕНЕРАЦИЯ” . | |
|  | (наименование профильной организации) | |

Обучающегося Костюнин Владислав Сергеевич,\_АСУб-20-2,\_\_\_\_

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

(ФИО, группа, подпись)

Руководитель практики от ИТиАД \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_

Копайгородский Алексей Николаевич, доцент

(ФИО, должность, подпись)

Руководитель практики организации \_ Шакиров Николай Шагеевич,

Инженер-электроник 1 категории

(ФИО, должность, подпись)

Оценка по практике

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

(ФИО, подпись, дата)

Содержание отчета на \_\_\_\_\_\_ стр. Приложение к отчету на \_\_\_\_\_ стр.

**Иркутск, 2022 г.**

**Содержание**

1. [Введение 5](file:///C:\Users\stud\Downloads\пример%20содержания.docx#_Toc94709035)

1.1 [Цель и задачи практики 5](file:///C:\Users\stud\Downloads\пример%20содержания.docx#_Toc94709035)

[2 Основная часть 5](file:///C:\Users\stud\Downloads\пример%20содержания.docx#_Toc94709036)

[1.1 Описание «ГУСИНООЗЕРСКАЯ ГРЭС» АО «ИНТЕР РАОЭЛЕКТРОГЕНЕРАЦИЯ» 6](file:///C:\Users\stud\Downloads\пример%20содержания.docx#_Toc94709037)

[1.2 Основные технические характеристики типа устройства 7](file:///C:\Users\stud\Downloads\пример%20содержания.docx#_Toc94709038)

[3 Индивидуальное задание 8](file:///C:\Users\stud\Downloads\пример%20содержания.docx#_Toc94709040)

[2.1 Диагностика неисправности конкретного устройства 8](file:///C:\Users\stud\Downloads\пример%20содержания.docx#_Toc94709041)

[2.2 Восстановление работоспособности конкретного устройства 10](file:///C:\Users\stud\Downloads\пример%20содержания.docx#_Toc94709042)

[3 Технологическая часть 14](file:///C:\Users\stud\Downloads\пример%20содержания.docx#_Toc94709043)

[3.1 Проверка работоспособности 14](file:///C:\Users\stud\Downloads\пример%20содержания.docx#_Toc94709044)

[Заключение 15](file:///C:\Users\stud\Downloads\пример%20содержания.docx#_Toc94709045)

[Список использованных источников 16](file:///C:\Users\stud\Downloads\пример%20содержания.docx#_Toc94709046)

**Введение**

**Цели и задачи прохождения практики:**

1. Осуществление работы инженера компьютерного программного обеспечения, разработка программного обеспечения, совершенствование практических навыков по всему циклу технологии разработки по месту прохождения практики;
2. участие в практической работе на месте прохождения производственной практики;
3. Получение практического опыта:
   1. Планирование и проектирование автоматизированных систем
   2. В Разработке программного обеспечения для спроектированной автоматизированной системы.
   3. Тестирование, выявление ошибок и недочетов
4. Разработка программного обеспечения
   1. Проектирование
   2. Разработка
      1. Дизайна
      2. Кодирование
      3. Тестирование
      4. Документирование
   3. Запуск приложения
5. Оформление отчета производственной практики

**Планируемые результаты практики**:

* отчет о похождение производственной практики;
* навыки разработки программного обеспечения, тестирование и отладки, написания документации

**Место прохождения:** ФИЛИАЛ «ГУСИНООЗЕРСКАЯ ГРЭС» АО «ИНТЕР РАО-ЭЛЕКТРОГЕНЕРАЦИЯ»

**Адрес:** 671160, РЕСПУБЛИКА БУРЯТИЯ, Г. ГУСИНООЗЕРСК, ГУСИНООЗЕРСКАЯ ГРЭС, ПРОМ-ПЛОЩАДКА ГРЭС

**2. Основная часть**

**2.1 Описание «ГУСИНООЗЕРСКАЯ ГРЭС»**

**АО «ИНТЕР РАО-ЭЛЕКТРОГЕНЕРАЦИЯ»**

Гусиноозерская ГРЭС является крупнейшей в Забайкалье электростанцией конденсационного типа и одним из крупнейших предприятий Республики Бурятия. За годы деятельности Гусиноозерская ГРЭС выработала более 160 млрд. кВт\*ч электроэнергии. Станция обеспечивает электроэнергией потребителей Бурятии и соседних регионов, а также тепловой энергией город Гусиноозёрск с населением более 24 тыс. чел. Выработка электроэнергии на станции составляет около 4,5 млрд. кВт\*ч в год. Отпуск тепла – около 300 тыс. Гкал.  
  
Четыре энергоблока первой очереди введены в эксплуатацию с 1976 по 1979 гг. Ввод второй очереди станции начался в 1988 г. запуском энергоблока № 5. Шестой энергоблок введен в эксплуатацию в 1992 г. В 1993–1996 гг. была проведена реконструкция энергоблока № 2. В конце октября 2013 года после реконструкции и технического перевооружения введен в эксплуатацию энергоблок № 4 мощностью 210 МВт.  
  
В ближайшие годы планируется провести работы по техническому перевооружению и реконструкции оборудования станции, что необходимо для повышения эффективности его работы и снижения себестоимости выпускаемой продукции.

С конца 90-х годов электростанция является градообразующим предприятием г. Гусиноозёрска.

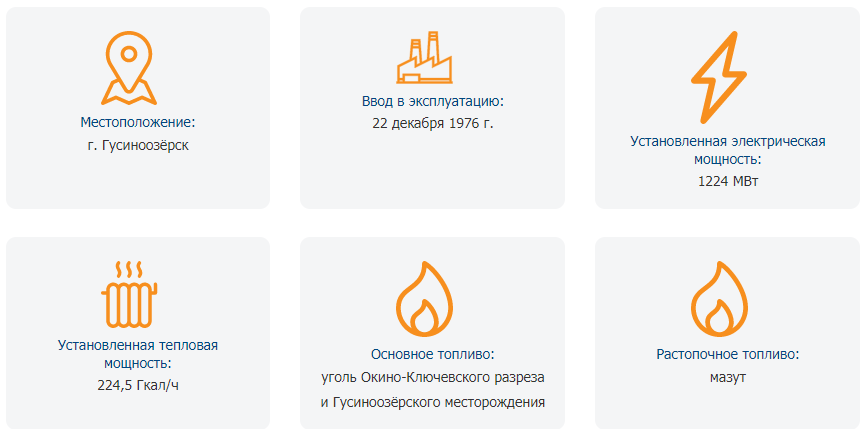
****

Рисунок 1 – Основные характеристики ГРЭС

Техническая архитектура на предприятии «ГУСИНООЗЕРСКАЯ ГРЭС» (рисунок 2).

**3. Индивидуальное задание**

Разработать WEB – приложение для организации процессов приема заявок на ремонт, автоматическое создание актов ремонта оборудования, системы хранения и заказа электронных компонентов, предназначенных для ремонта оборудования.

**3.1 Проектирование**

**3.1.1. Выбор технологий разработки WEB-приложения**



**Spring framework -** представляет собой просто контейнер внедрения зависимостей, с несколькими удобными слоями (например: доступ к базе данных, прокси, аспектно-ориентированное программирование, RPC, веб-инфраструктура MVC). Это все позволяет быстрее и удобнее создавать Java-приложения.



**Рисунок – слои Spring Framework**

В приложении будет использованы слои Data Access/Integration (ORM-Object Relation Mapping) для подключения Базы данных к приложению, слой WEB для создания MVC (Model-View-Controller) структуры общения сервера с клиентом приложения.

База данных приложения будет реализована благодаря:



**MySQL -** это реляционная система управления базами данных (СУБД), которая распространяется как свободное программное обеспечение. Является одной из наиболее популярных, так как отличается гибкостью, легкостью, удобством в использовании.



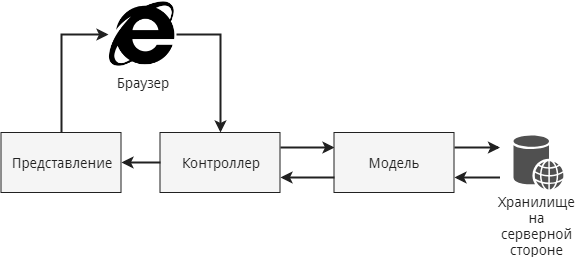
**Hibernate** — библиотека для языка программирования Java, предназначенная для решения задач объектно-реляционного отображения (ORM), самая популярная реализация спецификации JPA. Распространяется свободно на условиях GNU Lesser General Public License.

Позволяет сократить объёмы низкоуровневого программирования при работе с реляционными базами данных; может использоваться как в процессе проектирования системы классов и таблиц «с нуля», так и для работы с уже существующей базой.

Библиотека не только решает задачу связи классов Java с таблицами базы данных (и типов данных Java с типами данных SQL), но и также предоставляет средства для автоматической генерации и обновления набора таблиц, построения запросов и обработки полученных данных и может значительно уменьшить время разработки, которое обычно тратится на ручное написание SQL- и JDBC-кода. Hibernate автоматизирует генерацию SQL-запросов и освобождает разработчика от ручной обработки результирующего набора данных и преобразования объектов, максимально облегчая перенос (портирование) приложения на любые базы данных SQL.

**3.1.2 Выбор шаблона разработки веб-приложения**

**Model, View, Controller (MVC)** — шаблон (паттерн) программирования, разделяющий архитектуру приложения на три модуля: модель (Model), представление (View), контроллер (Controller). Он позволяет изменять каждый компонент независимо друг от друга для простой разработки и поддержки веб-приложений.



**3.1.2.1 Модели приложения**

**Модель** - данные, которые хранятся и обрабатываются на сервере.

В приложении будет реализовано 3 модели:

**Users** (Пользователи), **Components** (Компоненты), **Reports** (Документы)

Модели будет реализованы по принципу **CRUD**:

**Create** - создать

**Read** - прочесть

**Update** - редактировать

**Delete** – удалить

**3.1.2.2 Котроллеры приложения**

Контроллер обрабатывает входящие запросы. В фреймворке это может заключаться в определении конкретных URL, на которые попадает пользователь при переходе по ссылке или при нажатии кнопки.

Приложение будут реализованы контролеры:

* «/» - Главная страница
* «/login» - Страница входа
* «/registration» - Страница регистрации нового пользователя
* «/components» - Список электронных компонентов
* «/components/new» - Добавление нового компонента
* «components/update» - Редактирование компонента
* «/admin» - Страница управление пользователями
* «/reports» - Список всех заявок по ремонту
* «/reports/new» - Создать новую заявку

**2.3 Представления приложения**

**Представления (Вид**) — это HTML-шаблон, который возвращает сервер после обработки запроса. Если запрос корректно обрабатывается, вы получаете веб-страницу. Если запрос некорректный, вы попадаете на страницу ошибки 404.

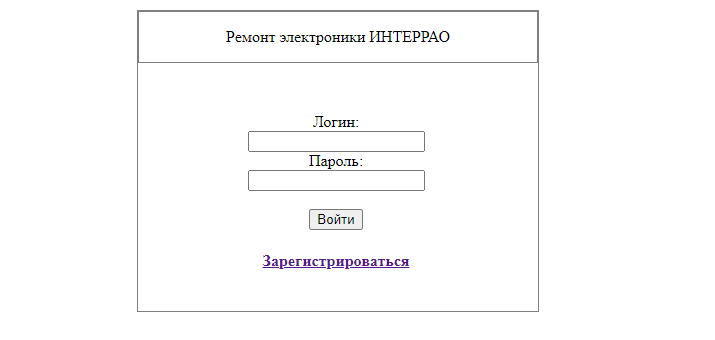


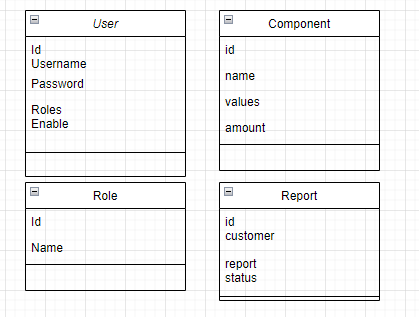
Рисунок – представление страницы входа

**3.2 Разработка**

В качестве средства разработки будем использовать **IDE IntelliJ IDEA** - интегрированная среда разработки программного обеспечения для многих языков программирования таких как JAVA, JavaSсript, HTML, CSS. Создадим проект при помощи сборщика проектов Apace Maven.

**3.2.1 Разработка Моделей**

Создадим 4 POJO(Plain old java object) объекта описывающие переменные у моделей.



К этим объектам создадим 4 службы реализующий интерфейс DAO(Data Access Object). Интерфейс DAO требует реализовать функции CRUD(Create, Read, Update, Delete) и служит отделением модели от бизнес процессов и для удобного подключения к базам данных.

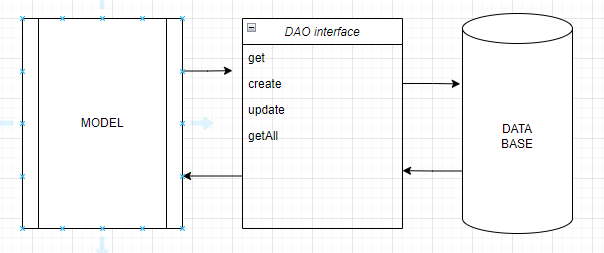
Службы:

UserService – служба пользователей

RoleService – служба пользовательских ролей

ReportService – служба заявок

ComponentsService – служба компонентов ремонта



**3.2.2 Разработка контроллеров**

Контролер будет вызывать методы из служб при получении HTTP запросов на заданный URL адрес. Основные типы HTTP запросов:

* GET — получение ресурса
* POST — создание ресурса
* PUT — обновление ресурса
* DELETE — удаление ресурса

Создадим 4 контроллера для каждой службы:

AdminController – Котроллер для работы с пользователями (Удаление, Добавление новой роли к пользователю)

ComponentController – Контроллер управление компонентами

ReportController - Контроллер управление заявками на ремонт

RegistrationController – Контроллер управления страницами регистрации и входа в приложение

**3.2.3 Разработка базы данных**