

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ  
Федеральное государственное автономное  
образовательное учреждение высшего образования  
«Южно-Уральский государственный университет  
(национальный исследовательский университет)»

Высшая школа электроники и компьютерных наук  
Кафедра «Электронные вычислительные машины»

РАБОТА ПРОВЕРЕНА

Рецензент

\_\_\_\_\_ 2020 г.  
«\_\_» \_\_\_\_\_

ДОПУСТИТЬ К ЗАЩИТЕ

Заведующий кафедрой ЭВМ

\_\_\_\_\_ Г.И. Радченко  
«\_\_» \_\_\_\_\_ 2020 г.

Разработка приложения для осуществления вибрационного контроля  
подшипников колесных пар железнодорожных вагонов

ПОЯСНИТЕЛЬНАЯ ЗАПИСКА  
К ВЫПУСКНОЙ КВАЛИФИКАЦИОННОЙ РАБОТЕ  
ЮУРГУ-090401.2020.653 ПЗ ВКР

Руководитель работы,  
к.т.н., доцент каф. ЭВМ  
\_\_\_\_\_ Д.В. Топольский  
«\_\_» \_\_\_\_\_ 2020 г.

Автор работы,  
студент группы КЭ-222  
\_\_\_\_\_ В.В. Игнатъев  
«\_\_» \_\_\_\_\_ 2020 г.

Нормоконтролёр,  
ст. преп. каф. ЭВМ  
\_\_\_\_\_ С.В. Сяськов  
«\_\_» \_\_\_\_\_ 2020 г.

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ  
Федеральное государственное автономное  
образовательное учреждение высшего образования  
«Южно-Уральский государственный университет  
(национальный исследовательский университет)»  
Высшая школа электроники и компьютерных наук  
Кафедра «Электронные вычислительные машины»

УТВЕРЖДАЮ  
Заведующий кафедрой ЭВМ  
\_\_\_\_\_ Г.И. Радченко  
« \_\_\_ » \_\_\_\_\_ 2020 г.

### **ЗАДАНИЕ**

**на магистерскую диссертацию**  
студенту группы КЭ-222  
Игнатъев Владимир Валерьевич  
обучающемуся по направлению  
09.04.01 «Информатика и вычислительная техника»

- 1. Тема работы:** «Разработка приложения для осуществления вибрационного контроля подшипников колесных пар железнодорожных вагонов» утверждена приказом по университету от 24 апреля 2020г. №627
- 2. Срок сдачи студентом законченной работы:** 1 июня 2020 г.
- 3. Исходные данные к работе:**
  - измерять уровень вибрации;
  - разработать протокол передачи данных по шине RS-485;
  - собрать конфигурацию рабочей станции, обеспечивающую достаточный уровень надежности обработки и хранения данных в условиях цеха (запыленность, вибрация, высокие температуры);

- работа с колесными парами моделей: РУ-Ш950, РУ1-Ш957 и т.д. На диагностику попадают колесные пары с пассажирских и грузовых железнодорожных вагонов;
- операционная система windows –это требование заказчика, т.к. им удобнее работать именно в данной операционной системе;
- приложение должно измерять уровень вибрации, строить график относительно частоты и силы вибрации, на основе которых оператор сможет сделать вывод о качестве колесной пары.

#### 4. Перечень подлежащих разработке вопросов:

- анализ предметной области;
- определение требований;
- проектирование;
- реализация;
- тестирование.

#### 5. Дата выдачи задания: 1 декабря 2020 г.

Руководитель работы:

Доцент каф. ЭВМ, к.т.н.  
должность, ученая степень

\_\_\_\_\_   
подпись

Топольский Д.В.  
ФИО ответственного

Студент \_\_\_\_\_

*/В.В. Игнатьев /*

## КАЛЕНДАРНЫЙ ПЛАН

Этап	Срок сдачи	Подпись руководителя
Введение и анализ предметной области	01.03.2020	
Определение требований	15.03.2020	
Проектирование	01.04.2020	
Реализация	01.05.2020	
Тестирование	15.05.2020	
Компоновка текста работы и сдача на нормоконтроль	24.05.2020	
Подготовка презентации и доклада	30.05.2020	

Руководитель работы \_\_\_\_\_ /Д.В. Топольский/

Студент \_\_\_\_\_ /В.В. Игнатьев/

## АННОТАЦИЯ

Игнатьев В. В. «Разработка приложения для осуществления вибрационного контроля подшипников колесных пар железнодорожных вагонов»: ФГАОУ ВО ЮУрГУ (НИУ), ВШ ЭКН; 2020, 76с. 16 ил., библиогр. список – 28 наим.

В рамках магистерской диссертации производится анализ технологий проведения вибрационной диагностики колесных пар железнодорожных вагонов. Организуется разработка приложения для осуществления вибрационного контроля подшипников колесных пар железнодорожных вагонов. Производится анализ результатов работы системы, её преимуществ и недостатков. Строится архитектура приложения.

Во введении рассматривается актуальность данной темы.

В первой главе производится анализ предметной области и обзор аналогов.

Во второй главе производится определение требований для данной системы, определение функциональных требований.

В третьей главе присутствует проектирование системы, ее архитектура и описание данных.

В четвертой главе показана реализация данной системы, ее интерфейсов.

В пятой главе производится тестирование системы.

В заключении описаны итоги проделанной работы.

Библиографический список показывает все дополнительные материалы и статьи, к которым было обращение при создании системы.

В приложении представлен исходный код системы.

## ОГЛАВЛЕНИЕ

ВВЕДЕНИЕ .....	7
1. АНАЛИЗ ПРЕДМЕТНОЙ ОБЛАСТИ.....	10
1.1. ОБЗОР АНАЛОГОВ.....	14
1.2 АНАЛИЗ ТЕХНОЛОГИЙ РАЗРАБОТКИ ВЕБ ПРИЛОЖЕНИЙ.....	19
1.2.1 Веб-приложения .....	19
1.2.2 СУБД.....	21
2. ОПРЕДЕЛЕНИЕ ТРЕБОВАНИЙ.....	29
2.1. ФУНКЦИОНАЛЬНЫЕ ТРЕБОВАНИЯ.....	29
3. ПРОЕКТИРОВАНИЕ .....	34
3.1. АРХИТЕКТУРА ПРЕДЛАГАЕМОГО РЕШЕНИЯ.....	34
3.2. ОПИСАНИЕ ДАННЫХ .....	37
4. РЕАЛИЗАЦИЯ .....	39
4.1. РЕАЛИЗАЦИЯ ПРИЛОЖЕНИЯ .....	39
5. ТЕСТИРОВАНИЕ .....	47
5.1. ТЕСТИРОВАНИЕ ПРИЛОЖЕНИЯ.....	47
6. ЗАКЛЮЧЕНИЕ .....	49
БИБЛИОГРАФИЧЕСКИЙ СПИСОК .....	51
ПРИЛОЖЕНИЕ А.....	54
ПРИЛОЖЕНИЕ Б .....	72

## ВВЕДЕНИЕ

В работе рассматривается тема разработки приложения для вибрационной диагностики подшипников колесных пар железнодорожных вагонов, это приложение должно помочь в проверке подшипников, которые находятся в колесных парах, без разборки самой колесной пары.

Одной из главных задач, стоящих перед тяговым подвижным составом (железнодорожные вагоны, локомотивы и т.д.) Российских железных дорог, является повышение их эксплуатационной надежности в целях обеспечения безопасности движения поездов с учетом реализации необходимых тяговых характеристик, улучшение габаритно-массовых и эксплуатационных показателей в широком диапазоне нагрузок и скоростей движения, в сложных климатических условиях .

В условиях эксплуатации железнодорожных вагонов подшипники букс колесных пар испытывают сочетание статических и динамических нагрузок. Существенную роль играют при этом неровности железнодорожного пути. На работу этих узлов влияют также возможные нарушения правил сборки буксового узла. Это приводит к образованию дефектов и отказов. Все большее значение приобретает необходимость контроля букс колесных пар путевых машин методами вибродиагностики, позволяющими проводить анализ технического состояния узла без его разборки.

Из опыта эксплуатации тягового подвижного состава известно, что при существующем методе эксплуатации и ремонта железнодорожной техники, когда замена узлов и механизмов происходит после отработки назначенного ресурса, не учитывается фактическое техническое состояние отработавших изделий. При этом значительная их часть имеет допустимое по ТУ техническое состояние, позволяющее производить дальнейшую эксплуатацию. В то же время, имеют место случаи преждевременного выхода элементов агрегатов и механизмов из строя. Повышение экономической

эффективности применяемого метода эксплуатации локомотивов и вагонов может быть достигнуто за счет внедрения нового, свободного от недостатков метода замены узлов и агрегатов по их фактическому техническому состоянию. Этот метод предусматривает после отработки агрегатами "гарантированного" ресурса, начиная с момента первой переборки, проведение непрерывного или периодического контроля и диагностики параметров, определяющих техническое состояние агрегатов с целью поддержания заданного уровня их надежности в эксплуатации на период до следующей проверки или ремонта при достижении ими предотказного состояния на основе диагностики.

Традиционные методы технического обслуживания путевых машин, применяемые в эксплуатации на железных дорогах, при производстве и ремонте, можно разделить на две категории: эксплуатация оборудования до выхода его из строя и планово-профилактическое обслуживание (по календарным срокам или по состоянию). Повышение технического уровня, качества и надежности машин, улучшение их использования во многом зависит от применяемых средств технической диагностики.

В современных средствах мониторинга и диагностики вращающегося оборудования основной вид анализируемых процессов - по параметрам вибрации, который активно вытесняет другие процессы, в том числе и тепловые. Вибрационная диагностика эффективнее и имеется тенденция к снижению затрат на ее реализацию. Применить мониторинг и диагностику по параметрам вибрации можно в любое время, даже через несколько лет эксплуатации оборудования, когда затраты на профилактические работы и ремонт превысят экономически оправданную величину. Поэтому многие фирмы переходят на обслуживание оборудования по состоянию узлов и машин.



Целью данной выпускной квалификационной работы—это разработать приложение для осуществления вибрационного контроля подшипников колесных пар железнодорожных вагонов.

Для достижения поставленной цели, необходимо решить следующие поставленные задачи:

1. Рассмотреть литературные источники, которые относятся к данной теме.
2. Провести обзор аналогов стендов для вибрационного контроля подшипников колесных пар железнодорожных вагонов.
3. Разработать приложение для тестирования колесных пар с помощью методов вибродиагностики.

В главе 1 рассказывается о предметной области разработки мобильных приложений, производится обзор существующих решений, для каждого аналога.

Из полученных данных в главе 1 будут сформулированы требования к приложению для производства вибрационной диагностики и детально описаны в главе 2, там же будет представлена диаграмма прецедентов.

Третья глава работы посвящена проектированию приложения. В ней будет описана архитектура, а также будет представлена структура базы данных приложения.

В глава 4 будет подробно рассказано о реализации приложения, будут представлены скриншоты приложения.

Глава 5 посвящена тестированию и определению эффективности работы мобильного приложения.

Источником информации были книги по разработке приложений, по проведению вибрационной диагностики.

# 1. АНАЛИЗ ПРЕДМЕТНОЙ ОБЛАСТИ

Прежде чем говорить о разработке приложения, необходимо понять, что из себя представляет вибродиагностика подшипников колесных пар и как, с помощью вибрации, можно определить качество подшипника.

Прежде всего вибродиагностика решает задачи по выявлению причин повышенного уровня вибрации агрегатов и обнаружению скрытых дефектов конструкций. Другими словами, вибродиагностика – это метод неразрушающего контроля машин, механизмов, узлов и агрегатов[1].

## **Неразрушающий контроль**

Неразрушающий контроль – это метод профессиональной дефектоскопии и диагностики без механического вмешательства в узлы и агрегаты. Позволяет выявить очаги возникновения неисправностей и отклонений от допустимого состояния узлов, агрегатов, сосудов и иных объектов.

Неразрушающий контроль используется для регулярной диагностики оборудования на опасных производственных объектах. Кроме того, эта форма контроля применяется в случаях, когда демонтаж и/или разборка агрегатов является трудоемким и дорогостоящим процессом. Неразрушающий контроль незаменим для проведения диагностики на действующем оборудовании.

## **Виды вибродиагностики**

Непрерывная диагностика или вибромониторинг. Данный вид вибродиагностики используется для ответственных узлов и/или дорогостоящих машин. Мониторинг обеспечивает непрерывный контроль за состоянием машины и сигнализацию в случае превышения предельно допустимого уровня вибрации. Непрерывная вибродиагностика позволяет[3]:

1. Построить график изменения уровня вибрации в зависимости от времени наработки.
2. Спрогнозировать время остаточной наработки агрегата до планового ремонта.
3. Избежать поломки агрегата из-за резкого возрастания вибрации.

Периодическая вибродиагностика. Применяется для диагностики машин, не требующих непрерывного виброконтроля. Периодическая вибродиагностика позволяет[4]:

1. Выявить причины повышенной вибрации, когда вибрации становится ощутимой, но еще не привела к поломке механизма.
2. Оценить соответствие уровня вибрации установленным нормам.
3. Определить возможность дальнейшей эксплуатации или способов модернизации «проблемных» агрегатов.

А сама вибродиагностика используется, для обеспечения безопасности железнодорожного движения предъявляет высокие требования к надежности работы ходовой части подвижного состава, и в первую очередь, к колесным парам и буксовым подшипникам вагонов. В случае локомотивов и моторных вагонов высокие требования к надежности работы дополнительно распространяются на подшипники приводов колесных пар, моторно-осевые подшипники и зубчатые зацепления привода КМБ и КРБ надо сначала расшифровать и только потом использовать сокращения. Вибродиагностика колесных пар вагонов проводится стандартными методами на стендах

вибродиагностики подшипников буксовых узлов колесных пар, также достоверные результаты о техническом состоянии подшипников получают тепловизионным контролем букс при движении в составе поезда.

### **Методы вибродиагностики буксовых узлов**

Источниками возникновения вибраций в буксовых подшипниках качения могут быть:

1. Нарушения формы поверхностей качения.
2. Раковины и трещины на поверхностях качения.
3. Недостаток/избыток смазки, разрыв масляного слоя.
4. Дефекты сепаратора.
5. Проскальзывание колец.
6. Задевания движущихся элементов о неподвижные.

Методы диагностирования основаны на анализе вибрации, создаваемой в подшипниках качения силами трения. Одним из методов поиска дефектов, нашедших широкое применение, является частотный (спектральный) анализ вибрации, позволяющий разделить её на компоненты разной частоты, возбуждаемые разными источниками колебательных сил и имеющих разную природу и разные свойства.

Наличие дефекта в подшипнике качения при их вращении приводит к появлению в вибросигнале интенсивных спектральных составляющих, положение которых на частотной оси зависит от места расположения дефекта, скорости вращения и геометрических размеров диагностируемых узлов. Амплитуда этих спектральных составляющих определяет степень развития того или иного дефекта. Таким образом, частотный анализ спектра вибросигнала позволяет определять как сам дефект, так и место его расположения. Следует отметить, что различные методы выявления дефектов в разной степени чувствительны к различным степеням развития дефекта в

контролируемом объекте. Например, спектральный анализ, как правило, хорошо начинает выявлять дефекты на средней их стадии развития[6].

Программа диагностики автоматически, зная частоту вращения, находит все характерные частоты элементов подшипника (сепаратора, роликов, колец), определяет амплитуду данных гармоник и запоминает ее. После набора необходимой статистической выборки она определяет и считает среднеквадратичное отклонение (СКО). Среднеквадратичное отклонение, используется для определения меры разброса случайной величины.

## 1.1. ОБЗОР АНАЛОГОВ

### Стенд для вибродиагностики: ОМСД-03

Основные технические характеристики	
Тип колесной пары	РУ1-950 РУ1Ш-957
Частота вращения, об./мин	~280
Время диагностики, мин.	1-1.5
Время обкатки, мин.	5
Система диагностики	МПП-93
Установленная электрическая мощность, кВт	5



Рисунок 1 – Стенд для вибродиагностики подшипников колесных пар железнодорожных вагонов ОМСД-03

Система вибродиагностики колесных пар на стенде позволяет производить обкатку колесной пары и выявляет дефекты подшипников буксовых узлов. Система имеет узел осевого нагружения, что позволяет диагностировать колёсные пары не только с роликовыми подшипниками, но и с подшипниками кассетного типа. Система выявляет следующие неисправности[7]:

1. Дефекты сборки подшипника.
2. Повреждения сепаратора.
3. Дефекты тел качения.
4. Износ тел качения.
5. Дефекты наружного и внутреннего кольца.
6. Повреждения поверхностей катания.

### **Описание системы управления**

ПКДМ - сбор и хранение диагностической информации; выдачу результатов диагностики в формате ГОДЕН/БРАК; •получение сводных отчетов за установленный период и развернутых по выбранному объекту; •передачу информации о проведенных замерах по СПД ОАО «РЖД» или через Интернет. Есть возможность измерения температуры буксовых узлов, что позволяет использовать бесконтактные пирометры, результаты которых учитываются в диагностики

## Стенд вибродиагностики подшипников колесных пар вагонов: СВ-01МВ

Основные технические характеристики	
Частота вращения, об./мин	~240
Время диагностики, с.	10
Время обкатки, мин.	5
Система диагностики	МПП-93
Установленная электрическая мощность, кВт	5,5

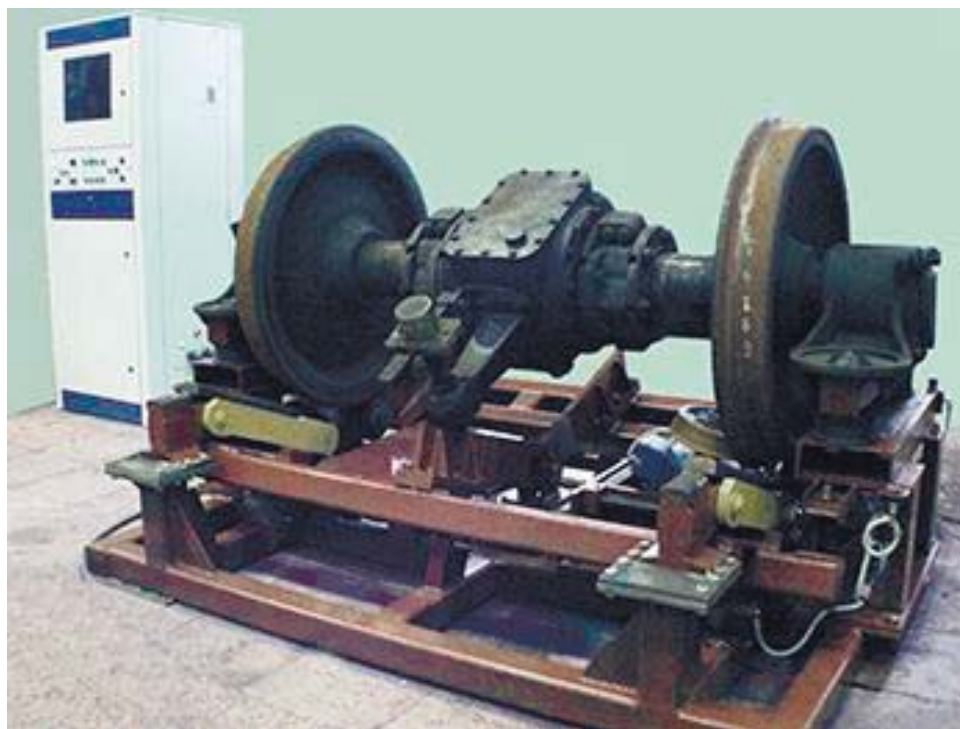


Рисунок 2 – Стенд для вибродиагностики подшипников колесных пар железнодорожных вагонов СВ–01 МВ

Данный стенд предназначен для определения технического состояния и выявления неисправностей подшипников (колец, роликов, сепараторов) буксовых узлов и их обкатки в соответствии с инструкцией ЗЦВРК[8].



## Описание системы управления

МПП-93 предназначена для анализа акустических шумов букс с целью их диагностирования при промежуточной ревизии букс.

Используется МПП-93 в комплекте со стендом вращения колесной пары СВ-01 МВ, поставляемым ФГУП «ЦНИИ «Буревестник» г.Нижний Новгород, ООО "Техинтел", г.Дзержинск или аналогичным. Возможно использовать стенд вращения колесной пары от установки УДП-85. Без стенда вращения колесной пары использование МПП-93 невозможно[8].



Рисунок 3 – Стенд для вибродиагностики подшипников колесных пар железнодорожных вагонов СВ–01 МВ

Автоматизированная установка МПП-93 предназначена для выявления неисправностей роликовых подшипников колесных пар без демонтажа буксовых узлов при промежуточной ревизии (в соответствии с инструктивными указаниями по эксплуатации и ремонту вагонных букс с роликовыми подшипниками ЗЦВРК).

Система МПП-93 позволяет выявлять следующие неисправности подшипников:

1. Задиры на торцах роликов и бортах внутренних колец подшипников; после точки пишут с заглавной буквы. Прошу исправить во всем тексте ПЗ.
2. Усталостные и коррозионные раковины на дорожках качения наружных и внутренних колец, а также роликов.
3. Разрывы внутреннего кольца.
4. Сколы борта наружного кольца.
5. Сколы торцов роликов.
6. Задиры и ползуны на дорожках качения внутреннего кольца и на поверхности ролика.
7. Изломы упорного кольца и сепаратора.

Вышеперечисленные неисправности могут быть выявлены только в том случае, если при вращении колесной пары от неисправных элементов буксового узла возникают дополнительные виброускорения, величина которых превышает на 10 % уровень собственных виброускорений колесной пары.

Автоматизированная установка МПП-93 дополняется возможностью создания базы данных всех проверяемых колесных пар. В базе данных содержатся: табельный номер дефектоскописта, номер проверенной колесной пары, результат проверки (брак выявлен или брак не выявлен), дата проверки, время проверки. Проверка акустических шумов производится по обеим сторонам колесной пары одновременно[8].

## 1.2 АНАЛИЗ ТЕХНОЛОГИЙ РАЗРАБОТКИ ВЕБ ПРИЛОЖЕНИЙ

### 1.2.1 Веб-приложения

Разработка web-приложения состоит из множества этапов, различие в стоимости сайта будет зависеть от количества времени, которое необходимо потратить инженеру-программисту для реализации определенной функциональности. На сегодняшний день у разработчиков web-сайтов есть богатый выбор относительно того, какой язык (или технологию) использовать для создания сайта. Вариантов много: Perl, PHP, ASP, ASP.NET, JSP, Coldfusion и т. д. Самыми распространенными (с большим отрывом от остальных) сегодня являются PHP и ASP.NET [9].

PHP (Hypertext Preprocessor) - язык для написания серверных сценариев (скриптов). Интерпретатор языка бесплатен, с открытым исходным кодом, созданы версии для различных веб-серверов, прежде всего, для Apache и Internet Information Services [10].

Преимуществами PHP являются: быстрая кастомизация готового решения (например, наложение нового дизайна на существующий сайт, который управляется CMS); большое количество бесплатных решений на этой платформе; абсолютная бесплатность; простота языка; любые данные в языке могут неявно быть текстовыми.

ASP.NET (Active Server Pages .Net) - средство для разработки веб-приложений от Microsoft. Asp.net - это не язык программирования, это технология, включающая в себя множество компонент [11].

К преимуществам ASP.NET можно отнести: наличие средства визуального программирования - уменьшение времени разработки в 2 и более раз; реализация объектной модели приложения; выполняемый код - откомпилированное приложение; поддержка технологии Microsoft.net - встроена в ОС Microsoft; встроенная поддержка технологии AJAX - любой

элемент формы может быть выполнен в «классическом» native-варианте или с помощью AJAX [12].

Для экспериментального исследования эффективности технологий разработки необходимо создать сайт и оценить, какая технология позволила максимально достичь поставленных целей при этом затратив минимальное количество времени и усилий для реализации проекта. Объектом исследования будет являться сайт автошколы «Диалог-сервис». К требованиям сайта будут относиться: возможность взаимодействия администрации со своими курсантами, реализация программ удаленного тестирования курсантов, предоставление информации о занятиях, индивидуальная запись на вождение транспорта, распределение курсантов на группы и составление расписания занятий для преподавателей.

Таким образом после реализации проекта необходимо провести анализ процесса разработки и выявить, посредством какой технологии получилось достичь максимального результата, и оценить время, за которое этот результат был достигнут.

## 1.2.2 СУБД

### MySQL

Одной из самых популярных СУБД на сегодняшний день является MySQL, распространяемая свободно (с некоторыми ограничениями). Эта серверная система способна эффективно функционировать во взаимодействии с интернет-сайтами и веб-приложениями. При этом она проста в освоении, что лишь увеличивает ее популярность[13].

Примечательно, что результатом огромной популярности MySQL стало появление в интернете множества руководств по освоению системы, а также огромного количества всевозможных плагинов и расширений, упрощающих работу с этой системой. Это в свою очередь лишь придало системе еще большей популярности[14].

Несмотря на отсутствие некоторого функционала, имеющегося у других СУБД, MySQL обладает достаточно обширным разнообразием доступных инструментов для создания приложений.

#### **Типы данных[15]:**

Символьные типы:

- CHAR: представляет строку фиксированной длины;
- VARCHAR: представляет строку переменной длины;
- TINYTEXT: представляет текст длиной до 255 байт;
- TEXT: представляет текст длиной до 65 КБ;
- MEDIUMTEXT: представляет текст длиной до 16 МБ;
- LARGETEXT: представляет текст длиной до 4 ГБ.

Числовые типы:

- TINYINT: представляет целые числа от -127 до 128, занимает 1 байт;
- TINYINT UNSIGNED: представляет целые числа от 0 до 255, занимает 1 байт;

- SMALLINT: представляет целые числа от -32768 до 32767, занимает 2 байта;
- SMALLINT UNSIGNED: представляет целые числа от 0 до 65535, занимает 2 байта;
- MEDIUMINT: представляет целые числа от -8388608 до 8388607, занимает 3 байта;
- MEDIUMINT UNSIGNED: представляет целые числа от 0 до 16777215, занимает 3 байта;
- INT: представляет целые числа от -2147483648 до 2147483647, занимает 4 байта;
- INT UNSIGNED: представляет целые числа от 0 до 4294967295, занимает 4 байта;
- BIGINT: представляет целые числа от -9 223 372 036 854 775 808 до 9 223 372 036 854 775 807, занимает 8 байт;
- BIGINT UNSIGNED: представляет целые числа от 0 до 18 446 744 073 709 551 615, занимает 8 байт;
- DECIMAL: хранит числа с фиксированной точностью. Данный тип может принимать два параметра precision и scale: DECIMAL(precision, scale).

### **Преимущества MySQL:**

Помимо универсальности и распространенности СУБД MySQL обладает целым комплексом важных преимуществ перед другими системами[13]. В частности следует отметить такие качества как:

1. Простота в использовании. MySQL достаточно легко устанавливается, а наличие множества плагинов и вспомогательных приложений упрощает работу с базами данных.

2. **Обширный функционал.** Система MySQL обладает практически всем необходимым инструментарием, который может понадобиться в реализации практически любого проекта.
3. **Безопасность.** Система изначально создана таким образом, что множество встроенных функций безопасности в ней работают по умолчанию.
4. **Масштабируемость.** Являясь весьма универсальной СУБД, MySQL в равной степени легко может быть использована для работы и с малыми, и с большими объемами данных.
5. **Скорость.** Высокая производительность системы обеспечивается за счет упрощения некоторых используемых в ней стандартов.

### **Недостатки MySQL:**

Как и любой программный продукт, система MySQL имеет определенные ограничения в своем функционале, что не позволяет использовать ее для работы с приложениями, имеющими некоторые специфические требования. К недостаткам этой СУБД относятся[17]:

1. **Недостаточная надежность.** В вопросах надежности некоторых процессов по работе с данными (например, связь, транзакции, аудит) MySQL уступает некоторым другим СУБД.
2. **Низкая скорость разработки.** Как и многим другим программным продуктам с открытым кодом, MySQL не достает некоторого технического совершенства, что порой сказывается на эффективности процессов разработки.
3. **Известные ограничения** – по задумке в MySQL заложены некоторые ограничения функционала, которые иногда необходимы в особо требовательных приложениях.

## SQLite

Легко встраиваемая в приложения база данных. Так как это система базируется на файлах, то она предоставляет довольно широкий набор инструментов для работы с ней, по сравнению с сетевыми СУБД. При работе с этой СУБД обращения происходят напрямую к файлам (в этих файлах хранятся данные), вместо портов и сокетов в сетевых СУБД. Именно поэтому SQLite очень быстрая, а также мощная благодаря технологиям обслуживающих библиотек[18].

### Типы данных SQLite

- null - значение NULL;
- integer - знаковое целочисленное значение, использует 1, 2, 3, 4, 6, или 8 байт в зависимости от порядка числа;
- real - число с плавающей точкой, занимает 8 байт для хранения числа в формате IEEE;
- text - текстовая строка, при хранении используются кодировки UTF-8, UTF-16BE или UTF-16LE;
- blob - тип данных BLOB, массив двоичных данных (предназначенный, в первую очередь, для хранения изображений, аудио и видео).

### Преимущества SQLite:

- файловая структура - вся база данных состоит из одного файла, поэтому её очень легко переносить на разные машины;
- используемые стандарты - хотя может показаться, что эта СУБД примитивная, но она использует SQL. Некоторые особенности опущены (RIGHT OUTER JOIN или FOR EACH STATEMENT), но основные все-таки поддерживаются;
- отличная при разработке и тестировании - в процессе разработки приложений часто появляется необходимость масштабирования.



SQLite предлагает всё что необходимо для этих целей, так как состоит всего из одного файла и библиотеки написанной на языке C[18].

### **Недостатки SQLite:**

- отсутствие системы пользователей - более крупные СУБД включают в свой состав системы управления правами доступа пользователей. Обычно применения этой функции не так критично, так как эта СУБД используется в небольших приложениях;
- отсутствие возможности увеличения производительности - опять, исходя из проектирования, довольно сложно выжать что-то более производительное из этой СУБД.

### **PostgreSQL:**

PostgreSQL является самым профессиональным из всех трех рассмотренных нами СУБД. Она свободно распространяемая и максимально соответствует стандартам SQL. PostgreSQL или Postgres стараются полностью применять ANSI/ISO SQL стандарты своевременно с выходом новых версий[13].

От других СУБД PostgreSQL отличается поддержкой востребованного объектно-ориентированного и/или реляционного подхода к базам данных. Например, полная поддержка надежных транзакций, т.е. атомарность, последовательность, изоляционность, прочность (Atomicity, Consistency, Isolation, Durability (ACID).) Благодаря мощным технологиям Postgre очень производительна. Параллельность достигнута не за счет блокировки операций чтения, а благодаря реализации управления многовариантным параллелизмом (MVCC), что также обеспечивает соответствие ACID. PostgreSQL очень легко расширять своими процедурами, которые называются **хранимые процедуры**. Эти функции упрощают использование постоянно повторяемых операций.

Хотя PostgreSQL и не может похвастаться большой популярностью в отличие от MySQL, существует довольно большое число приложений, облегчающих работу с PostgreSQL, несмотря на всю мощь функционала. Сейчас довольно легко установить эту СУБД используя стандартные менеджеры пакетов операционных систем.[18]

### **Типы данных PostgreSQL:**

- macaddr - MAC адрес;
- money - валютное значение;
- numeric [(p, s)] - точное численное значение с выбранной точностью;
- path - геометрическая кривая на плоскости;
- point - геометрическая точка на плоскости;
- polygon - многоугольник на плоскости;
- real - число с плавающей точкой одинарной точности (4 байта);
- smallint - знаковое целочисленное значение (4 байта);
- serial - автоматически инкрементируемое целочисленное значение (4 байта);
- text - строковое значение переменной длины;
- time [(p)] [without time zone] - время суток (без часового пояса);
- time [(p)] with time zone - время суток (включая часовой пояс);
- timestamp [(p)] [without time zone] - дата и время (без часового пояса);
- timestamp [(p)] with time zone - дата и время (с часовым поясом);
- tsquery - текстовый поисковый запрос;
- tsvector - документ текстового поиска;
- txid\_snapshot - пользовательский снимок транзакции с ID;
- uuid - универсальный уникальный идентификатор;
- xml - XML данные;
- bigint - знаковое 8-ми битное целочисленное значение;

- bigserial - автоматически инкрементируемое 8-ми битное целочисленное значение;
- bit[(n)] - строка постоянной длины;
- bit varying [(n)] - строка переменной длины;
- boolean - булево значение (true/false) ;
- box - прямоугольник на плоскости;
- bytea - бинарные данные (массив байтов) ;
- character varying [(n)] - строковое значение переменной длины;
- character [(n)] - строковое значение постоянной длины;
- cidr - IPv4/IPv6 сетевой адрес;
- circle - круг на плоскости;
- date - календарная дата (год, месяц, день) ;
- double precision - число с плавающей точкой двойной точности(8 байт);
- inet - IPv4/IPv6 адрес хоста;
- integer - знаковое 4-ех байтовое целочисленное значение;
- interval [fields][(p)] - отрезок времени;
- line - бесконечная прямая на плоскости
- lseg - отрезок на плоскости

### **Достоинства PostgreSQL:**

- открытое ПО соответствующее стандарту SQL - PostgreSQL - бесплатное ПО с открытым исходным кодом. Эта СУБД является очень мощной системой;
- большое сообщество - существует довольно большое сообщество, в котором вы запросто найдёте ответы на свои вопросы;
- большое количество дополнений - несмотря на огромное количество встроенных функций, существует очень много дополнений, позволяющих разрабатывать данные для этой СУБД и управлять ими;

- расширения - существует возможность расширения функционала за счет сохранения своих процедур;
- объектность - PostgreSQL это не только реляционная СУБД, но также и объектно-ориентированная с поддержкой наследования и много другого.

### **Недостатки PostgreSQL**

- производительность - при простых операциях чтения PostgreSQL может значительно замедлить сервер и быть медленнее своих конкурентов, таких как MySQL;
- популярность - по своей природе, популярностью эта СУБД похвастаться не может, хотя и присутствует довольно большое сообщество;
- хостинг - в силу выше перечисленных факторов иногда довольно сложно найти хостинг с поддержкой этой СУБД.

## 2. ОПРЕДЕЛЕНИЕ ТРЕБОВАНИЙ

Для реализации данной системы потребуется следующий набор подсистем:

1. Приложение с графическим интерфейсом под систему windows.
2. Api - серверное приложение для работы с данными написанное на php.
3. MySQL - база данных, для хранения зарегистрированных пользователей и данных о диагностике колесных пар и ее нежелательно заменять, иначе придется переписывать большую часть серверного приложения.
4. Chrome т.к приложение написано на базе chrome application на Javascript и в случаи его замены придется переписывать практически все клиентское приложение.

### 2.1. ФУНКЦИОНАЛЬНЫЕ ТРЕБОВАНИЯ

В данном подразделе определим какой функционал будет в приложении. Для этого воспользуемся диаграммой прецедентов.

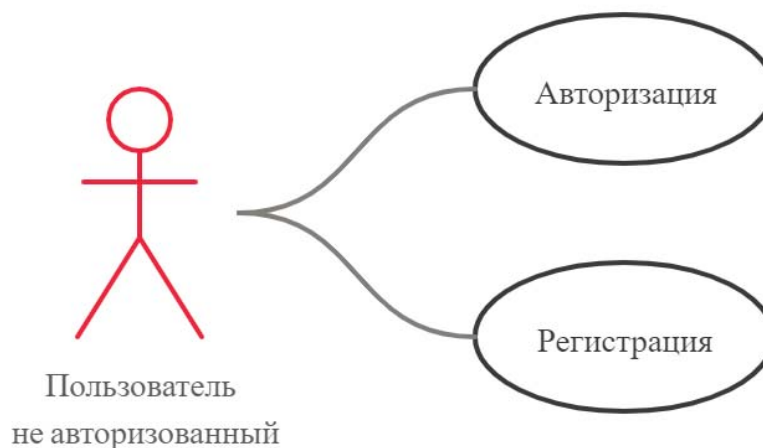


Рисунок 4 – Диаграмма вариантов использования приложения для не авторизованного пользователя

На рисунке 4 изображена диаграмма вариантов использования приложения для пользователя, которые еще не был авторизован в приложении. Данная диаграмма показывает, чтобы получить доступ к каким-либо функция приложения, необходимо иметь свою учетную запись в данном приложении.

1. Авторизация: первое, что требуется пользователю для начала работы в приложении - это авторизоваться в системе с помощью своей учетной записи. Как только пользователь авторизуется, приложение создает запрос через сервер к базе данных и проверяет о наличии данного пользователя с таким паролем. Если логин и пароль такие есть в базе, тогда приходит ответ о авторизации пользователя, если же такой пользователь не был обнаружен, тогда, приходит ответ о том, что "Логин или пароль введены неверно".
2. Регистрация: если же у пользователя нету учетной записи, тогда первым его шагом будет - это создать учетную запись, прежде чем он сможет приступить к работе в приложении. Когда пользователь нажимает кнопку регистрации, приложение открывает соответствующую форму, которую он должен заполнить. По заполнению которой, будет создан запрос, в результате исполнения которого, будет создана новая учетная запись в данном приложении с данным этого пользователя. После чего он сможет приступить к работе в данном приложении.

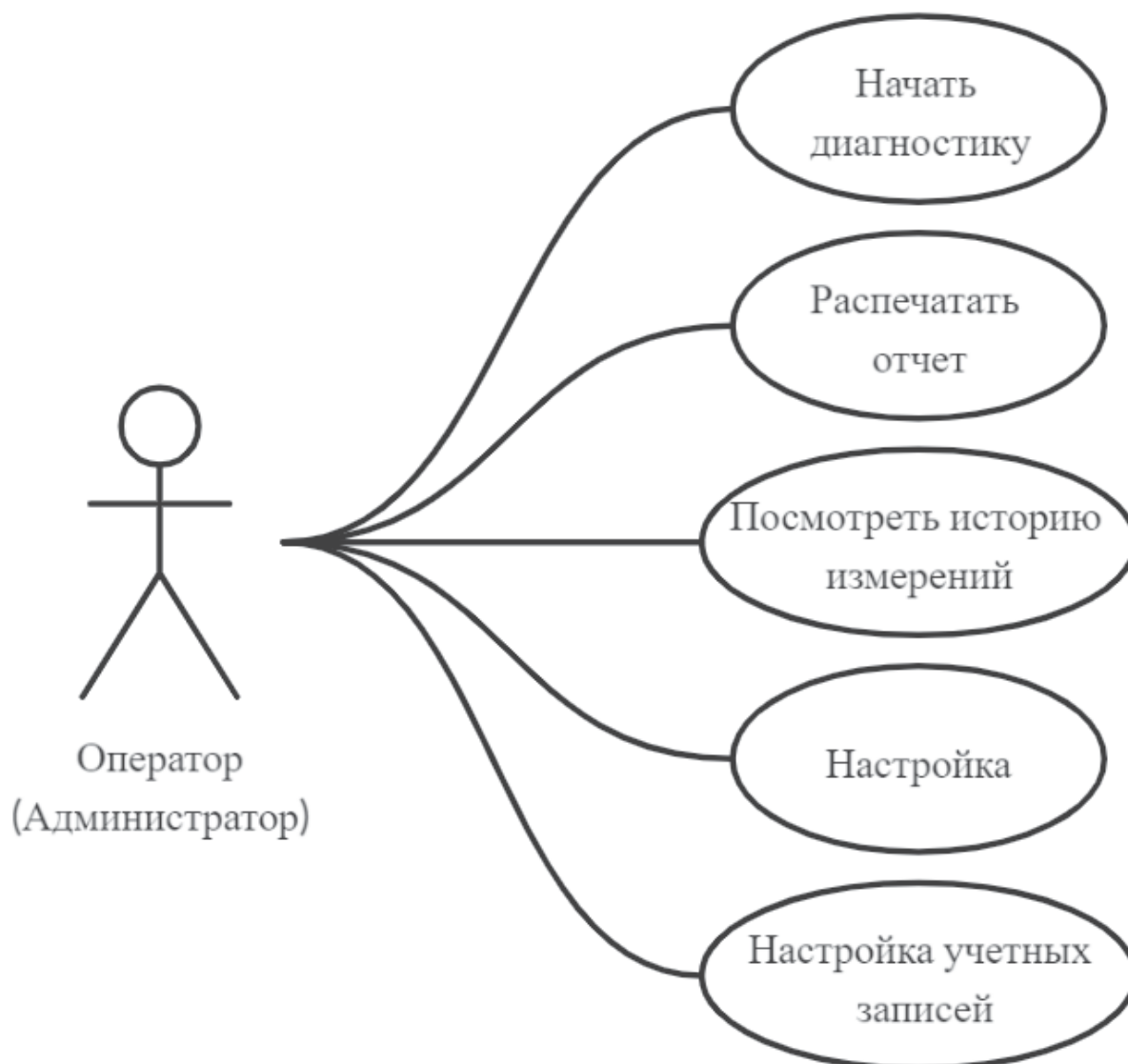


Рисунок 5 – Диаграмма вариантов использования приложения после авторизации в системе

На рисунке 5 изображена диаграмма вариантов использования приложения. Данная диаграмма показывает все возможности данного приложения с точки зрения простого Оператора и Администратора. Можно выделить следующие основных акторов, которые взаимодействуют с данной системой

1. Оператор(Администратор): пользователь приложения - это оператор, который проводит саму диагностику колесных пар или администратор, который отвечает за работоспособность системы, работу операторов и т.д. В зависимости от роли, пользователь наделяется разными правами доступа. Так, например, если у пользователя есть права администратора, он может не только посмотреть историю диагностики всех колесных пар, но и изменить или даже удалить любую из учетных записей, любого из операторов.
2. Начать диагностику: После авторизации, если авторизованный пользователь является Оператором, тогда он может приступить к калибровке колесных пар после того как настроит сам процесс калибровки. А настроить он это сможет в параметрах диагностики, которые могут быть заданы заранее.
3. Распечатать отчет: После завершения работы, оператор может распечатать весь отчет за весь рабочий день. В отчете будут указаны номер колесной пары, время, когда она была проверена, кто из операторов ее проверял, какие показатели получил оператор по завершении диагностики и бракованная или нет эта колесная пара. Но также можно получить все эти данные за любой период работы и любого из операторов.
4. Посмотреть историю измерений: допустим если в конце недели, потребуется посмотреть всю историю измерений, каждой колесной пары или каждого оператора, и после уже распечатать нужные данные за определенный период или определенного оператора, то это можно будет сделать в данной функции. Иначе говоря, сначала просматриваете историю всех диагностики, а после на основе этих данных формируется отчет и печатается.
5. Параметры диагностики: Прежде чем приступить к диагностике, оператор должен внести соответствующие параметры для нее. Которые он может получить, проведя диагностику на калибровочной колесной



паре, которая уже с дефектом в подшипнике. После чего он может приступить к калибровке других колесных пар, которые нуждаются в проверке.

6. Настройка учетных записей: когда пользователь авторизовался и у его учетной записи есть права администратора, тогда у него есть доступ к данной функции, которая позволяет не только полностью поменять данные учетной записи, но и полностью удалить учетную запись. А в изменяемые данные так же входит выдача или снятие прав администратора с любой учетной записи.

### 3. ПРОЕКТИРОВАНИЕ

#### 3.1. АРХИТЕКТУРА ПРЕДЛАГАЕМОГО РЕШЕНИЯ

При описании архитектуры разрабатываемого приложения была спроектирована общая диаграмма компонентов для отображения взаимодействия логических частей приложения. Которая представлена на рисунке 5.

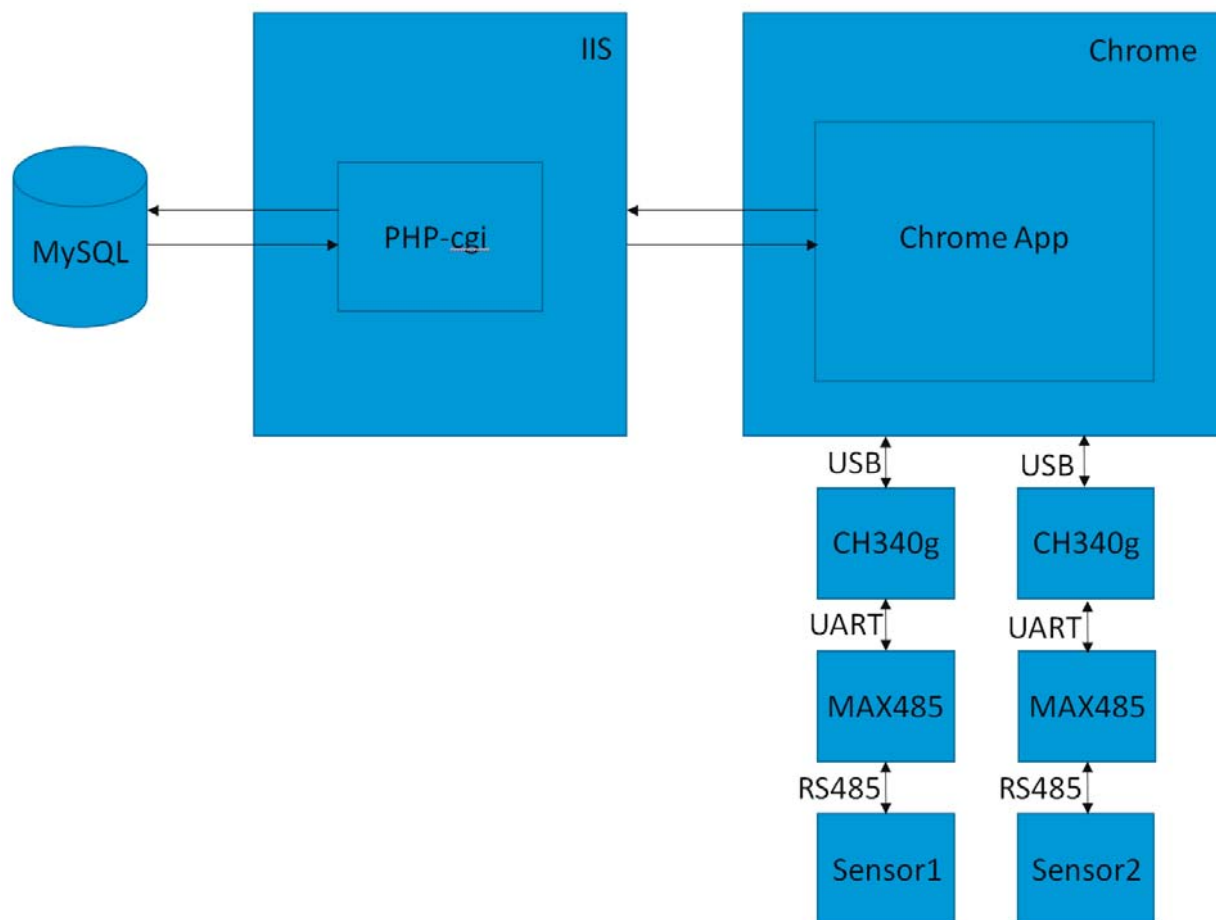


Рисунок 6 – Диаграмма компонентов проектируемой системы

MySQL - база данных, на которой хранятся все данные об измерениях и об учетных записях, всех операторов.

PHP-sgi - HTTP сервер с бэк-энд приложением на нем, которое обрабатывает данные и запросы от клиентского приложения и отправляет необходимые данные из базы данных (код приведен в Листинг Б.1):

- add user - функция отвечающая за регистрацию новых учетных записей;
- change user - функция отвечающая за смена учетной записи и за изменение данных учетной записи;
- delete user - функция отвечающая за удаление учетной записи;
- get measure chart - функция отвечающая за получение сырых данных, то есть без обработки;
- get measures - функция отвечающая за получение результатов измерения, после проведения диагностики;
- login - функция отвечающая за авторизацию пользователя;
- save measure - Сохранение измерений, полученные при проведении диагностики;
- user list - список всех учетных записей пользователей;
- auth - функция, которая отвечает за авторизацию пользователей, т.е которая проверяет о наличии той или иной учетной записи.

Chrome App - Клиентское веб-приложение, которое отвечает за получение данных о колесных парах при помощи двух датчиков, которые измеряют вибрацию и после получают среднеквадратичное отклонение, на основе которого и можно узнать исправен подшипник в или требуется ремонт (код приведен в Листинг А.1):

- login - функция, которая формирует запрос на сервер, на авторизацию пользователя;
- print - функция которая отвечает за печать сформированного отчета, о проведенных диагностиках подшипников колесных пар;
- start - функция отвечающая за начало получения данных с датчиков, которые измеряют вибрацию;
- disconnect - функция которая отвечает за отключение учетной записи, иначе говоря выход;

- `users` - функция которая открывает список пользователей, которые зарегистрированы в системе;
- `history.cancel` & `history.filter` - функции которые отвечают за применение определенного фильтра в истории всех проведенных диагностик, всех колесных пар;
- `save_measure` - функция отвечающая за сохранение данных, которые были получены после проведение диагностики колесной пары;
- `settings.cancel` - функция, отвечающая за отмену настроек, возвращая все параметры диагностики к значениям по умолчанию;
- `settings.start_calibration` - функция отвечающая за начало калибровки, это диагностика калибровочной колесной пары, которая заранее известно, что бракованная;
- `users.add` - функция которая формирует запрос на сервер, о добавлении нового пользователя, т.е. создании новой учетной записи;
- `users.modal.add`, `users.modal.change`, `users.modal.delete`- функции, которые отвечают за все операции, которые может проводить оператор с правами администратора, за изменение информации об учетной записи, за удаление учетной записи, зад добавления данных в учетную запись: например, выдача прав администратора.

### 3.2. ОПИСАНИЕ ДАННЫХ

Все данные в приложение принимаются и отправляются в формате JSON. Такой формат был выбран из-за удобства чтения и легковесности, нежели, например, XML.

Описание базы данных. В приложении есть своя локальная база данных, в которой храниться информация не только о всех учетных записях, но и о всех проведенных калибровках, всех колесных пар. И вся эта информация храниться в формате Json

Список таблиц и значений:

Таблица measurings:

- ID(Integer) – идентификатор;
- calibration (TinyInt)– данные полученные при диагностике калибровочной колесной пары;
- wheelset\_number (Integer)–номер колесной пары;
- operator\_id (Integer)– идентификатор оператора;
- operator\_opinion (Text)– мнение оператора, которое он выносит после диагностики колесной пары;
- date (BigInteger)– дата проведения диагностики;
- sensor1 (Json)– датчик №1;
- sensor2 (Json)– датчик №2;
- sensor1\_max\_sko (Integer) – максимальное значение СКО с сенсора №1;
- sensor2\_max\_sko (Integer) – максимальное значение СКО с сенсора №1;
- sensor1\_max (Integer) – максимальное значение сырых данных, без обработки с первого датчика;
- sensor2\_max (Integer) – максимальное значение сырых данных, без обработки со второго датчика;

- sensor1\_solution (TinyInt) – показывает состояние колесной пары на датчики №1;
- sensor2\_solution (TinyInt) – показывает состояние колесной пары на датчики №1.

Таблица session\_key:

- user (Integer) – данные учетной записи оператора, которые проводил диагностику одной из колесных пар;
- key(VarChar) – внешний ключи;
- auth\_date(TimeStamp) – дата и время сеанса диагностики.

Таблица users:

- ID (Integer)– идентификатор учетной записи;
- login (VarChar) – логин учетной записи;
- pass (VarChar) – пароль учетной записи;
- first\_name (VarChar) – имя оператора в учетной записи;
- middle\_name (VarChar) – отчество оператора в учетной записи;
- last\_name (VarChar) – фамилия оператора в учетной записи;
- admin (TinyInt) – наличие или отсутствие прав администратора;
- deleted (TinyInt) – удалена учетная запись или нет.

## 4. РЕАЛИЗАЦИЯ

### 4.1. РЕАЛИЗАЦИЯ ПРИЛОЖЕНИЯ

Приложение было реализовано в среде разработки Netbeans, это среда, разработанная с открытым исходным кодом для разработчиков программного обеспечения. Среда предоставляет все средства, необходимые для создания профессиональных десктоп приложений, корпоративных, мобильных и веб-приложений на платформе Java, а также C/C++, PHP, JavaScript, Groovy и Ruby.

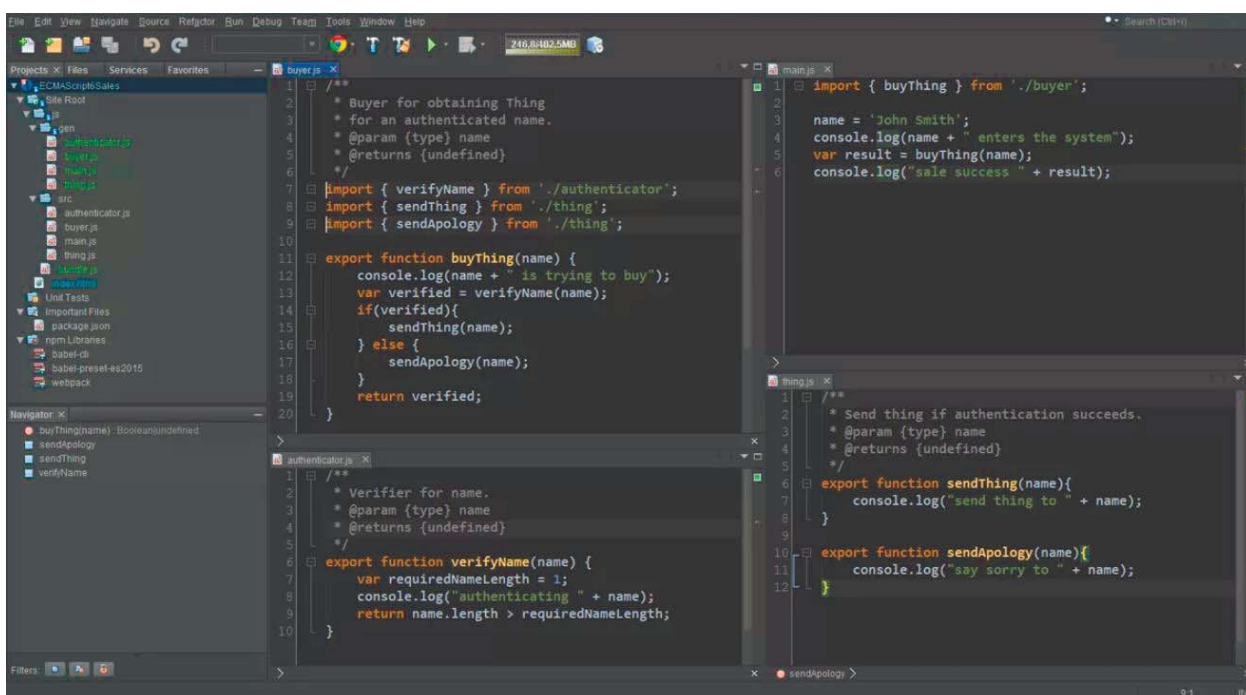


Рисунок 7 – Главный экран среды разработки Netbeans

Клиентское приложение было реализовано на языке javascript(js), а серверное приложение(backend) было реализовано на языке PHP. PHP – это серверный язык программирования, который выполняется на стороне сервера, в то время как JavaScript выполняется в браузере на стороне пользователя.

Javascript - поддерживает объектно-ориентированный, императивный и функциональный стили. Плюс, для нее создано большое количество

всевозможных фреймворков, заточенных под разные задачи и облегчающие процесс разработки.

РНР - методология программирования, основанная на представлении программы в виде совокупности объектов, каждый из которых является экземпляром определённого класса, а классы образуют иерархию наследования.

Идеологически ООП — подход к программированию как к моделированию информационных объектов, решающий на новом уровне основную задачу структурного программирования: структурирование информации с точки зрения управляемости, что существенно улучшает управляемость самим процессом моделирования, что, в свою очередь, особенно важно при реализации крупных проектов.

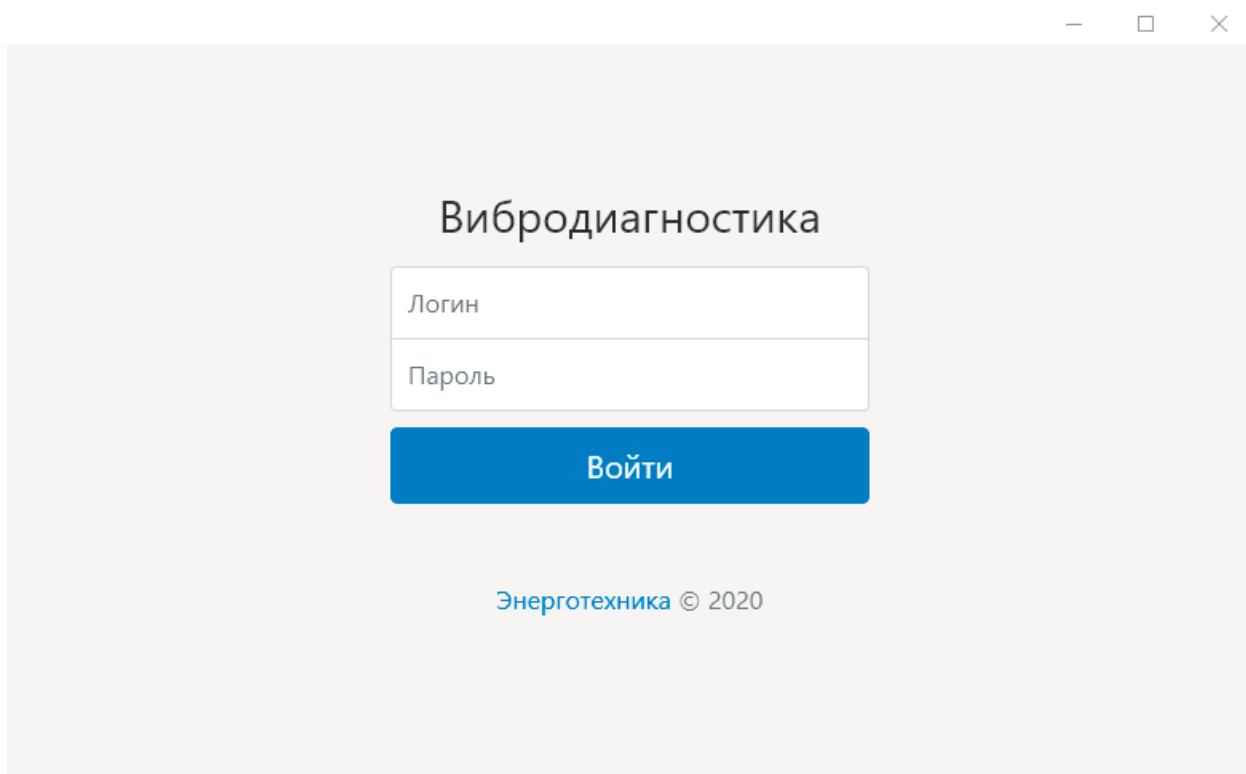


Рисунок 8 – Экран авторизации оператора.

После запуска приложения оператор может зайти в свою учетную запись и начать свою работу или найти необходимые данные и распечатать их.



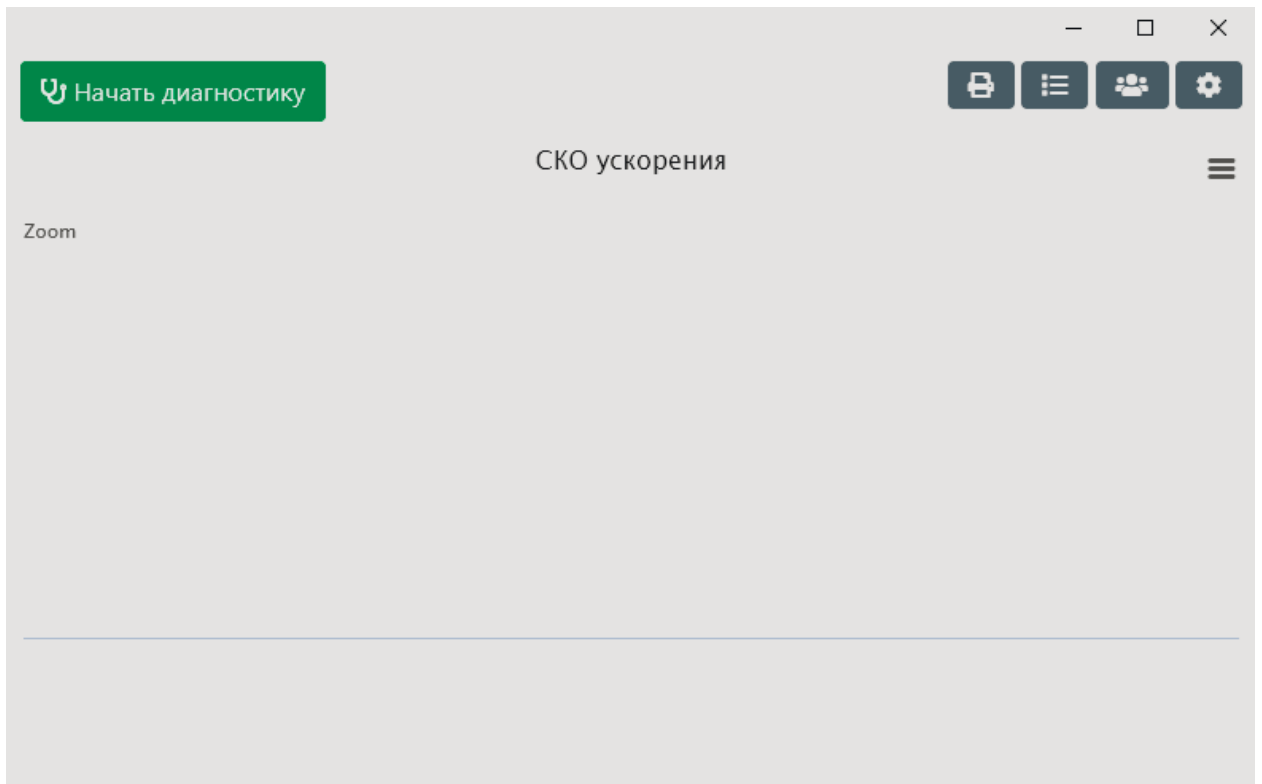


Рисунок 9 – Экран начала диагностики

После прохождения процесса авторизации, оператор может сразу начать диагностику, если настройки диагностики были заранее установлены

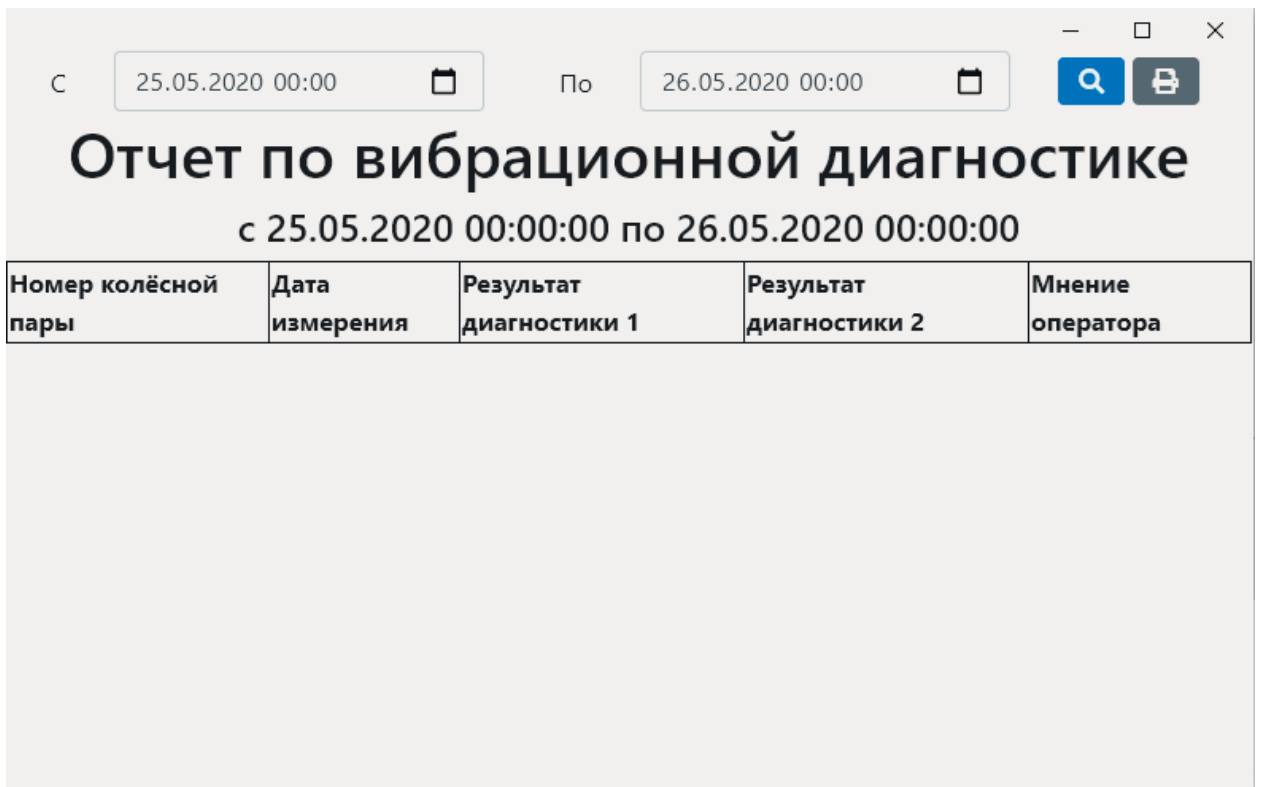


Рисунок 10 – Экран для печати отчета по проведенным работам

Оператор может распечатать отчет по проведенным диагностикам колесных пар, абсолютно за любой период. В отчете будет отражена вся информация о каждом процессе диагностике. В отчет будет входить:

1. Номер колесной пары.
2. Дата измерения.
3. Результат диагностики.
4. ФИО оператора, который проводил диагностику.
5. Мнение и заключение оператора.

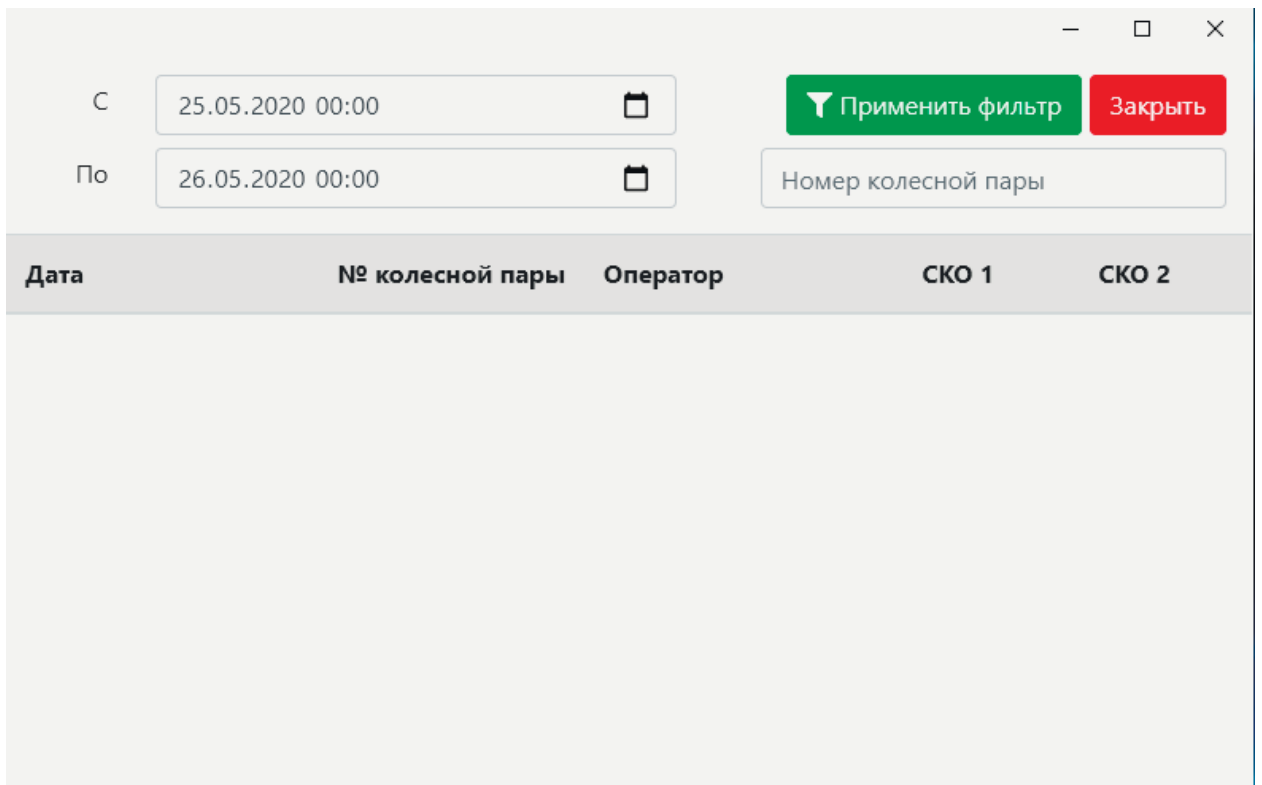


Рисунок 11 – Экран просмотра истории всех измерений

На данном экране показывается история всех измерений, проведенных в цехе. На основе информации из истории и формируется и определяется отчет. Иначе говоря, в истории можно узнать за какой период отчет нужно формировать для отчетности, например.



Рисунок 12 – Главный экран среды разработки Netbeans

Экран с учетными записями, в которых оператор с правами администратора, может не только изменять данные учетных записей, но и администрировать всю деятельность других операторов.

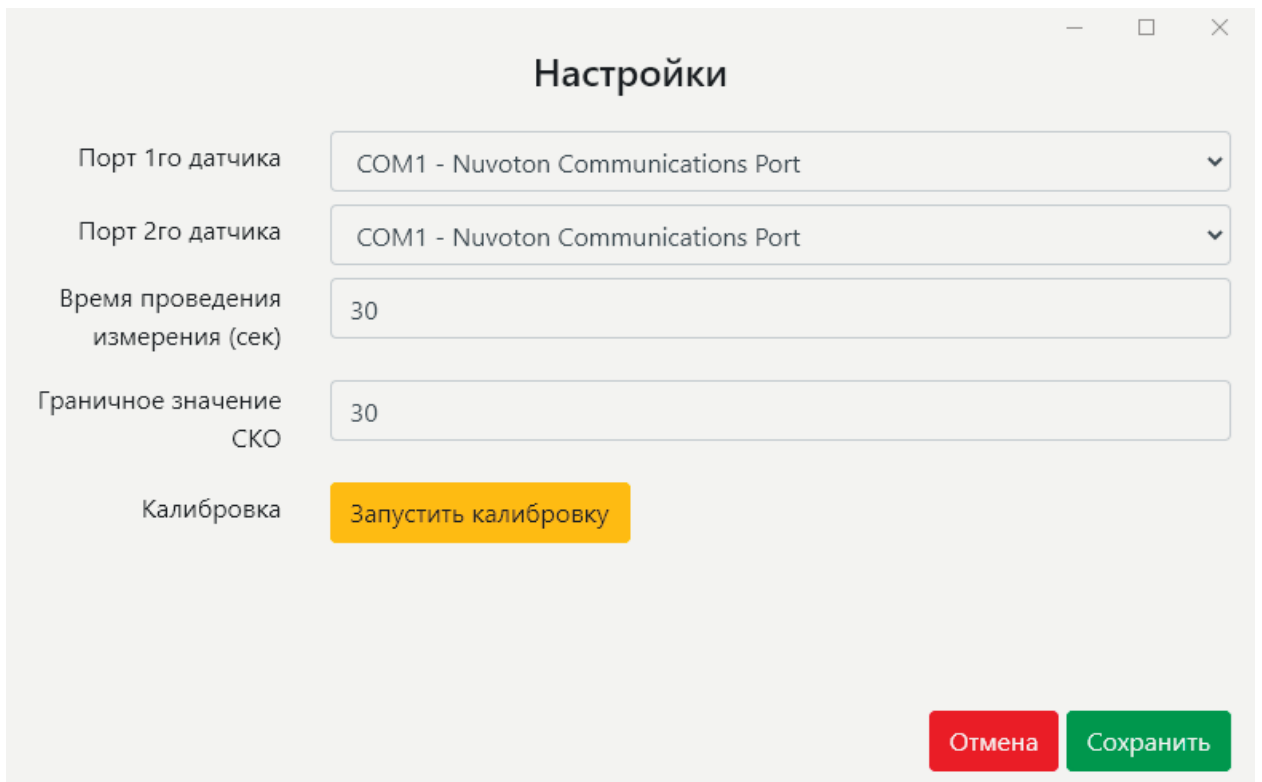


Рисунок 13 – Главный экран среды разработки Netbeans

На данном экране оператор настраивает датчики с которых будет принимать данные приложение и сам процесс диагностики колесной пары. Максимальное значение СКО(Среднеквадратичного отклонения) определяется на основе калибровочной колесной пары, которая уже с браком в конструкции.

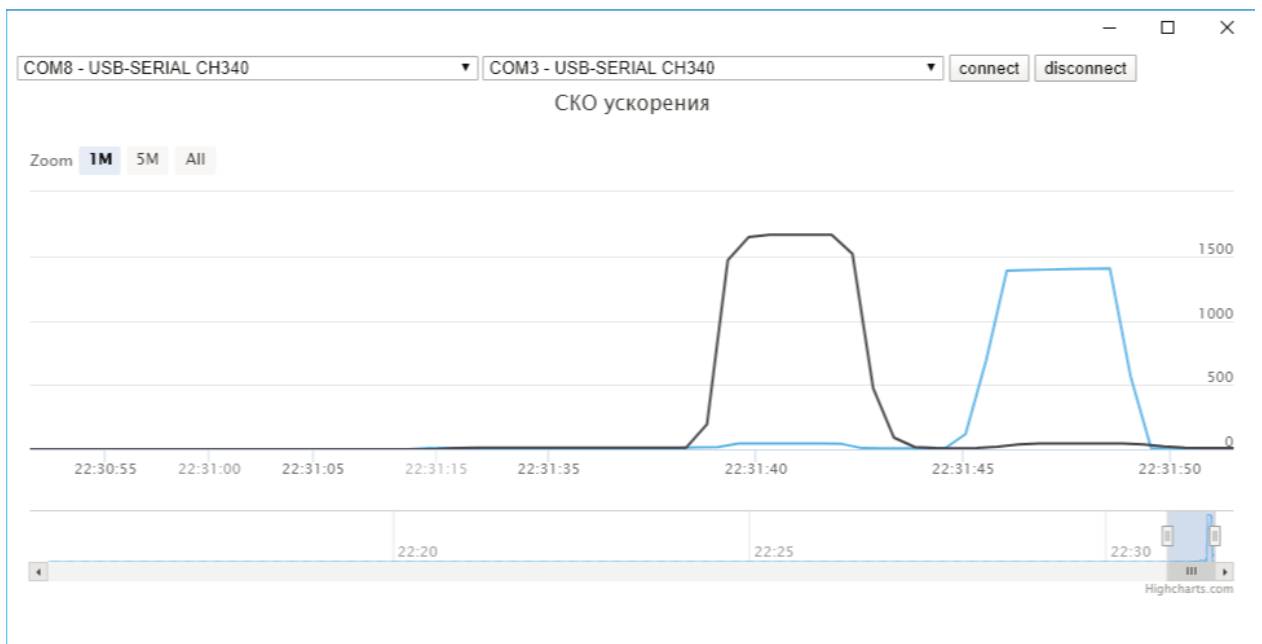


Рисунок 14 – Главный экран среды разработки Netbeans

Основной экран процесса диагностики, где можно увидеть частоты вибрации каждого подшипника, и если частота вибрации выше определенного уровня, то оператор может отправить ее на повторную диагностику, но уже другим оператором или сделать вывод, что колесная пара подлежит ремонту. Иначе говоря, бракованная.

## 5. ТЕСТИРОВАНИЕ

### 5.1. ТЕСТИРОВАНИЕ ПРИЛОЖЕНИЯ

Приложения для вибродиагностики подшипников колесных пар железнодорожных вагонов, было протестировано, как в лабораторных условиях, так и в полноценных рабочих условиях.

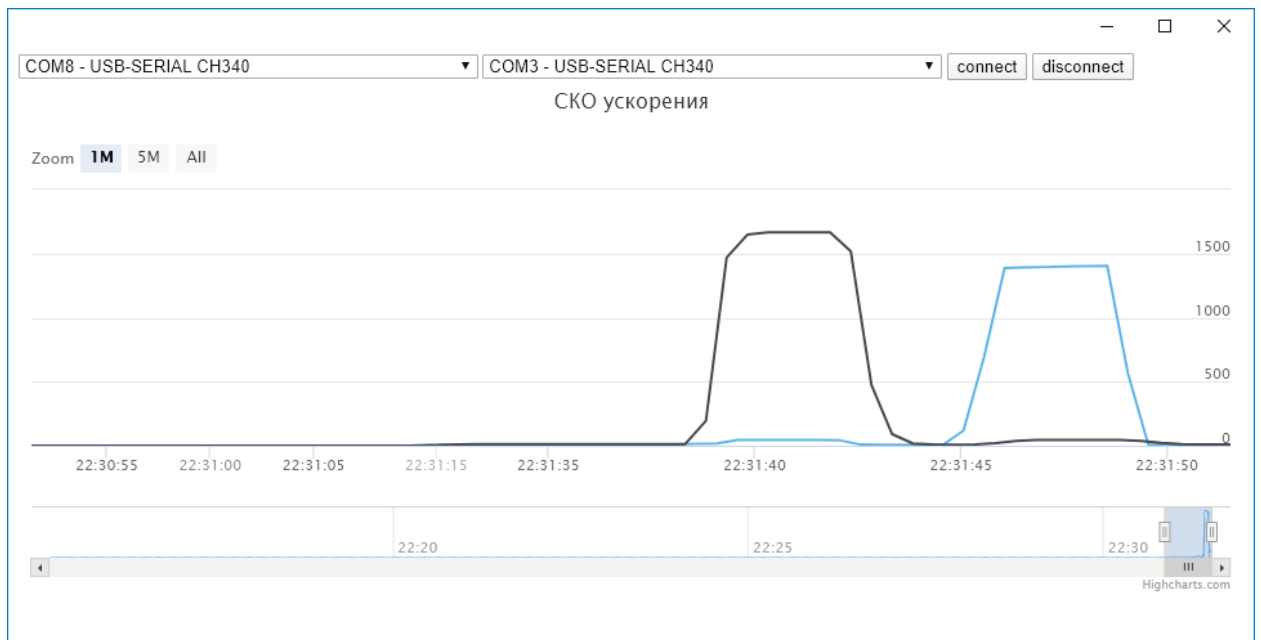


Рисунок 15 – Тестирование приложения для вибродиагностики подшипников колесных пар железнодорожных вагонов в лабораторных условиях

В рамках тестирования на колесных парах в реальном железнодорожном цеху. Были проведены диагностики нескольких колесных пар, включая калибровочную колесную пару, которые заранее имеет дефект в своей конструкции.

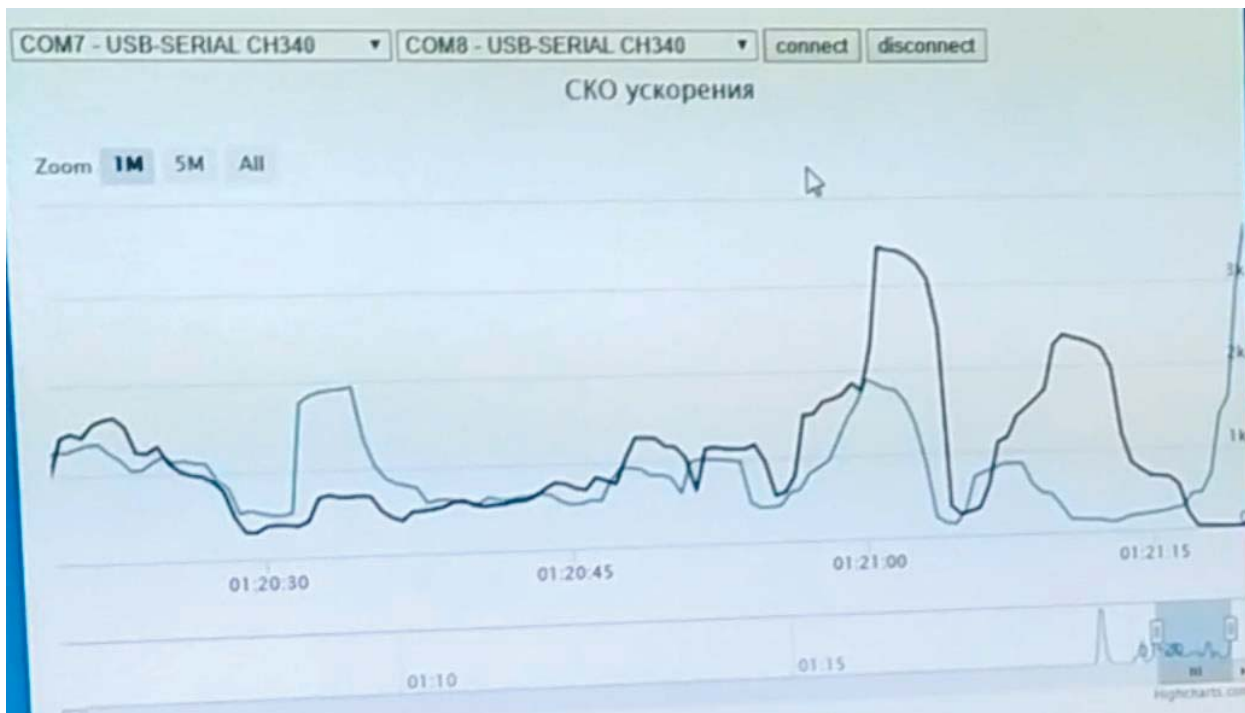


Рисунок 16 – Тестирование приложения для вибродиагностики подшипников колесных пар железнодорожных вагонов на реальных колесных парах.

На основе проведенных тестов и на основе высказанного мнения заказчика, можно сделать вывод, что приложение корректно отображает все, даже самые мелкие, колебания подшипника колесной пары.

Стоит отметить, что полной автоматизации этого процесса в рамках этой работы не удалось, из-за того, что тот стенд, для которого была проделана вся эта работы, не предусматривает своей конструкцией полной автоматизации, т.е. без участия человека проводить диагностику подшипников колесных пар железнодорожных вагонов.

Так же можно добавить, что для ручного тестирования использовались обычные сценарии взаимодействия.



## 6. ЗАКЛЮЧЕНИЕ

В представленной работе были разобраны все основные темы, которые связаны с проектированием, разработкой и вводом в эксплуатацию приложения для вибродиагностики подшипников колесных пар железнодорожных вагонов. Были выполнены следующие задачи:

1. Изучены материалы и опыт разработок: в области создания веб-приложений, проведения вибродиагностик, проверка подшипников на износ с помощью вибродиагностики.
2. Проанализированы технологии, которые будут полезны при разработке данного приложения и которые помогут упростить разработку его.
3. На основе полученных данных, было разработано и реализовано приложение для вибродиагностики подшипников колесных пар железнодорожных вагонов.
4. Были проанализированы полученные результаты и определена эффективность работы приложения для вибродиагностики подшипников колесных пар.

Реализация веб-приложения выполнена полностью и реализована под операционную систему Windows.

В результате проделанной работы можно сделать следующие выводы:

- построена архитектура приложения на основе chrome application;
- создано, реализовано и внедрено в эксплуатацию приложения для вибродиагностики подшипников колесных пары железнодорожных вагонов, которое полностью устроило заказчиков;
- организован удобный поиск и автоматическое формирование отчетов необходимых данных по диагностике подшипников колесных пар железнодорожных вагонов;

- было разработано бэк-энд приложение, для обработки полученных данных и для взаимодействия с базой данных.

Данное приложение имеет дружелюбный интерфейс и интуитивно понятное управление для операторов, которые будут на нем работать.

## БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. Азовцев, Ю.А. Баркава, Н.А. Гаузе, А.А. Вибрационная диагностика роторных машин и оборудования: уч. пособие/ СПб: СПб ГТУРП, 2014. – 127 с: ил. 77.
2. Петрухин, В.В. Основы вибродиагностики и средства измерения вибрации. Учебное пособие. Гриф УМО вузов России / В.В. Петрухин. – М.: Инфра-Инженерия, 2010. – 736 с.
3. Петрухин, С. В. Основы вибродиагностики и средства измерения вибрации / С.В. Петрухин. – М.: Инфра-Инженерия, 2010. – 449 с.
4. Костюков, В.Н. Науменко, А.П. Вибродиагностика поршневых компрессоров // Компрессорная техника и пневматика. – 2002. – № 3. – С. 30-31 ;
5. ГОСТ 32106-2013. Контроль состояния и диагностика машин. Мониторинг состояния оборудования опасных производств. Вибрация центробежных насосных и компрессорных агрегатов. – М.: Стандарт-Информ, 2013. – 8-10 с.;
6. Колобов, А.Б. Вибродиагностика. Теория и практика – Изд-во «Инфра-Инженерия», 2020. – 252 с..
7. Комплекс ОМСД-03 буксовых узлов колесных пар грузовых и пассажирских вагонов [http://xn--d1apfn.xn--p1ai/doc/omsd-03\\_passazhirskih\\_gruzovih\\_kolesnih\\_par.pdf](http://xn--d1apfn.xn--p1ai/doc/omsd-03_passazhirskih_gruzovih_kolesnih_par.pdf) (Дата обращения: 20.05.2020).
8. Установка для вибродиагностики буксовых узлов колесных пар железнодорожных вагонов автоматизированная МПП-93-Р <http://www.priborgt.ru/mpp93> (Дата обращения: 20.05.2020).
9. Низамутдинов Марсель, Разработка Web-приложений/ Марсель Низамутдинов. - М.: БХВ-Петербург, 2019. - 761 с.

10. Введение в разработку веб-приложений на PHP и MySQL – <https://webformymself.com/vvedenie-v-razrabotku-veb-prilozhenij-na-rnr-i-mysql/> (Дата обращения: 10.03.2020).
11. Борьба технологий: сайты на ASP.NET против сайтов на PHP [Электронный ресурс]. URL: [http://www.ti-net.ru/publikatsii/saity\\_na\\_aspnet/](http://www.ti-net.ru/publikatsii/saity_na_aspnet/) (дата обращения: 15.05.2020).
12. Сравнение разработки на ASP.NET и PHP [Электронный ресурс]. URL: <http://progresoft.ru/articles?fid=5&aid=127> (дата обращения: 16.05.2020).
13. Пирогов, В.Ю. Информационные системы и базы данных: организация и проектирование: Учебное пособие / В. Пирогов. – СПб.: ВHV, 2009. – 528 с.
14. Шнайдер, Р.Л. Microsoft SQL Server 6.5. Проектирование высокопроизводительных баз данных / Шнайдер, Роберт. – М.: ЛОРИ, 2014. – 361 с
15. Мартишин, С.А. Проектирование и реализация баз данных в СУБД MySQL с использованием MySQL Workbench: Методы и средства проектирования информационных систем и техноло / С.А. Мартишин, В.Л. Симонов, М.В. Храпченко. – М.: Форум, 2018. – 61 с.
16. Дубнов, П.Ю. Access 2000. Проектирование баз данных / П.Ю. Дубнов. – М.: ДМК, 2015. – 272 с
17. SQLite, MySQL и PostgreSQL: сравниваем популярные реляционные СУБД - <https://tproger.ru/translations/sqlite-mysql-postgresql-comparison/> (Дата обращения: 10.05.2020).
18. Машнин, Т.С. Google App Engine Java и Google Web Toolkit: разработка Web-приложений / Т.С. Машнин. - СПб.: ВHV, 2014. - 352 с.
19. Фридман, А. Л. Построение Интернет-приложений на языке php. Практический курс / А.Л. Фридман. – Москва: РГГУ, 2016. – 336 с.
20. Ленгсторф, Д.В. PHP и jQuery для профессионалов / Джейсон Ленгсторф. – М.: Вильямс, 2017. – 362 с.

21. Никсон, Р.П. Создаем динамические веб-сайты с помощью PHP, MySQL и JavaScript / Робин Никсон. – М.: Питер, 2013. – 496 с.
22. Колисниченко, Денис Профессиональное программирование на PHP (+CD-ROM) / Денис Колисниченко. – М.: БХВ-Петербург, 2015. – 416с.
23. Что такое XML и JSON. Их особенности – <https://pointschool.ru/что-такое-xml-i-json-ih-osobennosti/> (Дата обращения: 16.05.2020).
24. ГОСТ Р 53563-2009. Мониторинг оборудования опасных производств. Порядок организации. – М.: Стандартинформ, 2010. – 5 с.;
25. JSON основы использования <https://ruseller.com/lessons.php?id=1212> (Дата обращения: 10.03.2020).
26. Запись данных в формате – JSON <https://habr.com/ru/post/230079/> (Дата обращения: 10.03.2020).
27. JSON краткое руководство – <https://coderlessons.com/tutorials/java-tekhnologii/vyuchit-json/json-kratkoe-rukovodstvo>
28. Что такое XML. – <https://msiter.ru/tutorials/uchebnik-xml-dlya-nachinayushchih/что-такое-xml>. (Дата обращения: 18.05.2020).