

СОДЕРЖАНИЕ

СОДЕРЖАНИЕ.....	5
ВВЕДЕНИЕ.....	6
1. МЕТОДЫ И ИНСТРУМЕНТЫ ПРОГРАММНОЙ ИНЖЕНЕРИИ.....	8
1.1. Техническое задание.....	8
1.2. Анализ существующих аналогов.....	14
1.3. Перечень задач, подлежащих решению в процессе разработки.....	15
1.4. Обоснование выбора ОС для разработки.....	15
1.5. Обоснование выбора СУБД.....	17
1.6. Обоснование выбора языков программирования.....	20
1.7. Обоснование выбора сред разработки.....	22
1.8. Описание реализуемой архитектуры клиент-сервер.....	24
1.9. Выбор средства для взаимодействия с базой данных.....	24
1.10. Информация о QR-коде.....	25
Выводы.....	26
2. ПРОЕКТИРОВАНИЕ КОМПОНЕНТОВ ПРОГРАММНОГО ПРОДУКТА.....	28
2.1. Разработка структуры системы.....	28
2.2. Разработка базы данных.....	28
2.3. Построение диаграммы вариантов использования Windows приложения.....	31
2.4. Структура Android приложения.....	32
2.5. Организация сетевого взаимодействия.....	37
Выводы.....	40
3. ТЕСТИРОВАНИЕ И ИНТЕГРАЦИЯ КОМПОНЕНТОВ ПРОГРАММНОГО ПРОДУКТА.....	41
3.1. Тестирование системы.....	41
3.2. Руководство администратора.....	42
3.3. Руководство пользователя.....	43
Выводы.....	54
ЗАКЛЮЧЕНИЕ.....	55
СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ.....	56
Основная литература.....	56
Дополнительная литература.....	56

ВВЕДЕНИЕ

Актуальность темы курсового проекта обусловлена тем, что в настоящее время промышленные предприятия часто нуждаются в различных программных решениях, которые смогли бы увеличить производительность труда за счет качественного анализа, планирования имеющихся ресурсов и уменьшения времени поиска необходимой информации. Актуальным является создание системы, состоящей из компонентов, которые в совокупности будут решать необходимые задачи и как результат улучшать функционирование предприятия.

Объектом курсового проекта является промышленное производство.

Предметом исследования курсового проекта является автоматизация учета оборудования на промышленном предприятии

Целью проекта является разработка информационной системы, позволяющей вести учет производственных активов

Для достижения поставленной цели решаются следующие задачи:

1. Выполнить анализ предметной области
2. Провести сравнительный анализ существующих аналогов
3. Определить оптимальную структуру системы
4. Осуществить выбор средств реализации программного продукта, соответствующего выбранной структуре.
5. Реализовать базу данных и программные компоненты системы
6. Осуществить тестирование компонентов
7. Разработать сопроводительную документацию

1. МЕТОДЫ И ИНСТРУМЕНТЫ ПРОГРАММНОЙ ИНЖЕНЕРИИ

1.1. Техническое задание

Наименование системы

Настоящее Техническое задание определяет требования и порядок создания «Системы автоматизированного учета контрольно-измерительного оборудования в подразделении предприятия».

Основания для разработки

В связи с ростом парка измерительного оборудования на предприятиях ведение его бумажного учета и получение нужной информации становится трудоемкой задачей. Возникает необходимость в систематизации и сокращении времени работы по учету и контролю оборудования. Решение предполагает переход от бумажного вида работ к электронному. В рамках данной работы разрабатывается система для учета, контроля и перемещения измерительного оборудования в подразделении предприятия. Такая система позволит вести учет о наличии, срокам поверки и движении контрольно-измерительного оборудования в подразделении предприятия, а также предоставит оперативный подбор необходимого оборудования по его техническим характеристикам для создания измерительных стендов различного назначения. Кроме того, система предоставит оперативный доступ к информации о нужном оборудовании ограничивая местоположение сотрудника лишь пределами локальной сети предприятия.

Исполнитель

Исполнителем проекта является студент Калужского филиала МГТУ им. Н. Э. Баумана, факультета ИУК, группы ИУК4-51Б, Иванов Иван Иванович.

Краткая характеристика области применения

Разрабатываемая система предназначена для применения в области выполнения работ по учету стандартного измерительного оборудования в подразделении предприятия.

Целевая аудитория

Инженерно-технический персонал.

Назначение системы

Разрабатываемая система должна обеспечивать систематизированную работу по учету контрольно-измерительного оборудования и предоставлять оперативный доступ к нужной информации через локальную сеть предприятия.

Цели создания системы

Целью создания системы является усовершенствование системы контроля измерительного оборудования в подразделении предприятия.

Плановые сроки начала и окончания работы по созданию системы

Планируемые сроки начала и окончания работы над проектом:
09.09.2022 – 03.12.2022.

Требования к приложению на ПК

Общие сведения в формах ввода и отображения любого типа контрольно-измерительного оборудования должны содержать следующие поля:

Вид оборудования (выбирается):

- измерительное оборудование;
- индикаторный прибор;
- вспомогательное оборудование.

Тип оборудования (выбирается):

- Измерительный комплекс;
- Анализатор;
- Приемник;

- Осциллограф;
- Генератор;
- Антенна;
- Пробник;
- Токосъемник;
- Мультиметр;
- Источник питания;

Производитель (вводится пользователем);

Наименование (вводится пользователем);

Заводской номер (вводится пользователем);

Инвентарный номер (вводится пользователем);

Дата ввода в эксплуатацию (вводится пользователем в формате ДД.ММ.ГГГГ.);

Дата очередной поверки (вводится пользователем в формате ДД.ММ.ГГГГ.);

Для различных типов оборудования должны быть созданы формы отображения технических характеристик, значения которых составляют детализированную таблицу оборудования. Одна часть содержит общие сведения о контрольно-измерительном оборудовании, а вторая содержит технические характеристики этого оборудования.

Технические характеристики в формах ввода и отображения любого типа контрольно-измерительного оборудования уточняются на этапе проектирования.

Система должна содержать формы добавления нового оборудования. Поля этой формы должны учитывать типы характеристик, присущие добавляемому оборудованию. Должна быть предусмотрена возможность изменения характеристик и параметров уже введенного оборудования.

Система должна содержать средства, позволяющие производить поиск сведений об оборудовании по следующим признакам:

- по типу;

- по заводскому номеру;
- по инвентарному номеру;
- по срокам поверки.

Система должна выводить отчет.

Вся информация в отчете должна отображаться в двух видах:

- общая информация для всех приборов – в единой таблице,
- технические характеристики каждого типа приборов – в таблицах

оборудования.

Требования к мобильному приложению

Приложение должно обеспечивать следующий функционал:

- устанавливать соединение с удаленной базой данных локальной сети предприятия;
- предоставлять возможность автоматического фокусирования при считывании QR-кода;
- предоставлять возможность автоматической подсветки при недостаточной освещенности;
- производить считывание данных с QR-кода оборудования;
- используя полученные данные получать от сервера общую информацию о приборе и его технических характеристиках;
- выводить полученную информацию на экран;
- предоставлять возможность генерации QR кода и автоматического сохранения его в памяти телефона.

Тип зашифрованной информации в QR-коде и вид выводимой на экран информации уточняются на этапе проектирования.

Требования к надежности

Программный продукт должен устойчиво функционировать и обеспечивать надежную защиту данных.

Надежное (устойчивое) функционирование системы должно быть обеспечено выполнением пользователем (заказчиком) совокупности организационно-технических мероприятий, перечень которых приведен ниже:

- организацией бесперебойного питания технических средств;
- использованием лицензионного программного обеспечения;
- регулярным выполнением рекомендаций Минтруда РФ, изложенных в Постановлении от 23 июля 1998 г. №28 «Об утверждении Межотраслевых типовых норм времени на работы по сервисному обслуживанию персональных электронно-вычислительных машин и организационной техники и сопровождению программных средств»;
- регулярным выполнением требований ГОСТ 51188-98 (защита информации, испытание компьютера на наличие компьютерных вирусов);

Время восстановления после отказа

Время восстановления после отказа, вызванного сбоем электропитания технических средств (иными внешними факторами), не фатальным сбоем (не крахом) операционной системы, не должно превышать времени восстановления операционной системы и восстановления работы сети.

Время восстановления после отказа, вызванного неисправностью технических средств, фатальным сбоем (крахом) операционной системы, не должно превышать времени, требуемого на устранение неисправностей технических средств и переустановки программных средств.

Условия эксплуатации

Климатические условия эксплуатации, при которых должны обеспечиваться заданные характеристики, должны удовлетворять требованиям, предъявляемым к техническим средствам в части условий их эксплуатации.

Требования к квалификации и численности персонала

Система требует наличия пользователей двух категорий – рядовых пользователей, сотрудников предприятия, которым требуется вести учет контрольно-измерительного оборудования и своевременное получение необ-

ходимой информации об оборудовании и администраторов, настраивающих работу с удаленной базой данных, управляющих разграничением прав доступа для отдельных пользователей и устраняющих неисправности в случае сбоя.

Требования к защите информации и программ

Защита информации осуществляется разграничением прав доступа – рядовые пользователи не должны иметь возможность вносить изменения, которые могут повлиять на функционирование системы.

Требования к программной документации

Должны быть разработаны следующие программные документы:

1. Расчетно-пояснительная записка:

- Техническое задание;
- Научно-исследовательская часть;
- Проектная часть;
- Производственно-технологическая часть;
- Организационно-экономическая часть;
- Раздел охраны труда и экологии;

2. Графическая часть - 7 листов формата А1 включающие в себя:

- демонстрационные чертежи;
- структурные схемы;
- основные алгоритмы;
- ER-диаграмму.

Стадии разработки

Техническое задание

1. Обоснование перспективности реализуемого проекта:

- постановка задачи;
- сбор базовых материалов;
- установка критериев системы;
- необходимость проведения исследовательских работ;

2. Исследовательская работа:

- выбор оптимальных методов решения поставленной задачи;
- определение требований к техническим средствам;
- обоснование практической возможности реализации данного

проекта;

3. Разработка и утверждение технического задания:

- определение требований к проекту;
- определение стадий, этапов и сроков разработки проекта и доку-

ментации на нее;

- согласование и утверждение технического задания;

Технический проект

1. Разработка технического проекта:

- определение формы представления входных и выходных данных;
- определение конфигурации технических средств;

2. Утверждение технического проекта:

- установка плана по разработке проекта;
- создание пояснительной записки;
- утверждение технического проекта;

Этапы разработки

На стадии разработки технического задания должен быть выполнен этап разработки, согласования и утверждения настоящего технического задания.

На стадии проектирования программы должен быть выполнен этап выборки программного обеспечения, системы управления базой данных, библиотек для создания, этап проектирования системы в целом, разработка рабочей документации.

На стадии реализации производится разработка и тестирование спроектированной программы.

1.2. Анализ существующих аналогов

В настоящее время существуют аналоги предоставляющие подобный функционал. Однако данные системы являются дорогостоящими и в дополнение к актуальному перечню задач предоставляют множество других, которые зачастую являются избыточными и только нагружают, и усложняют систему, а также требуют больших аппаратных ресурсов.

1.3. Перечень задач, подлежащих решению в процессе разработки.

Для выполнения поставленной задачи необходимо определить оптимальную архитектуру для построения системы и средства реализации программного продукта, соответствующего данной архитектуре. Структура системы должна включать в себя сервер и набор клиентов. Серверная часть состоит из БД с информацией об оборудовании и приложения для работы с БД. Клиентская часть будет реализована в виде мобильного приложения, которое позволит получать данные от сервера, путем передачи ему информации, зашифрованной в QR коде оборудования и генерировать такие коды при необходимости. Средства реализации включают в себя: СУБД для реализации базы данных, языки программирования для реализации серверного и клиентского приложений, ОС для разработки, среды разработки, инструмент взаимодействия между клиентами и сервером базы данных.

1.4. Обоснование выбора ОС для разработки

Поскольку на заводских предприятиях персональные компьютеры в большей части находятся под управлением ОС Windows и к тому же используемые там инженерные программы адаптированы под данную ОС, то рассмотрение других систем для разработки серверной части системы не является необходимым.

Вопрос выбора мобильной ОС стоит рассмотреть более подробно. Существуют несколько операционных систем на которых функционируют мобильные устройства, среди них можно выделить: Android, iOS, Windows

Phone, BlackBerry. При выборе наиболее важный фактор – распространенность платформы среди пользователей. Учет данного факта повысит вероятность того, что при внедрении системы на предприятии уже с конкретной целевой аудиторией не возникнет проблем с наличием устройств под управлением ОС данного типа.

По данным онлайн сервиса statcounter.com, предоставляющего общемировую статистику использования web-браузеров и операционных систем, лидирующую позицию среди мобильных операционных систем за период с апреля 2017 года по март 2018 года занимает ОС Android - 73,3% (рисунок 2.1).

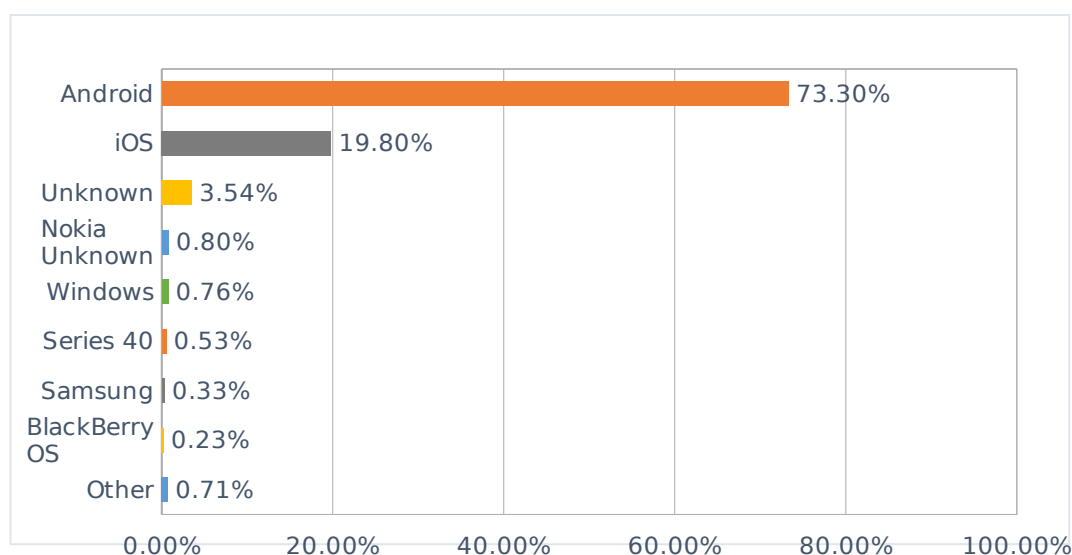


Рисунок 2.1. Рынок мобильных ОС (апрель 2017 г. – март 2018 г.)

Причем лидирующие позиции ОС Android сохраняет уже на протяжении многих лет и число пользователей продолжает стремительно расти (рисунок 2.2).

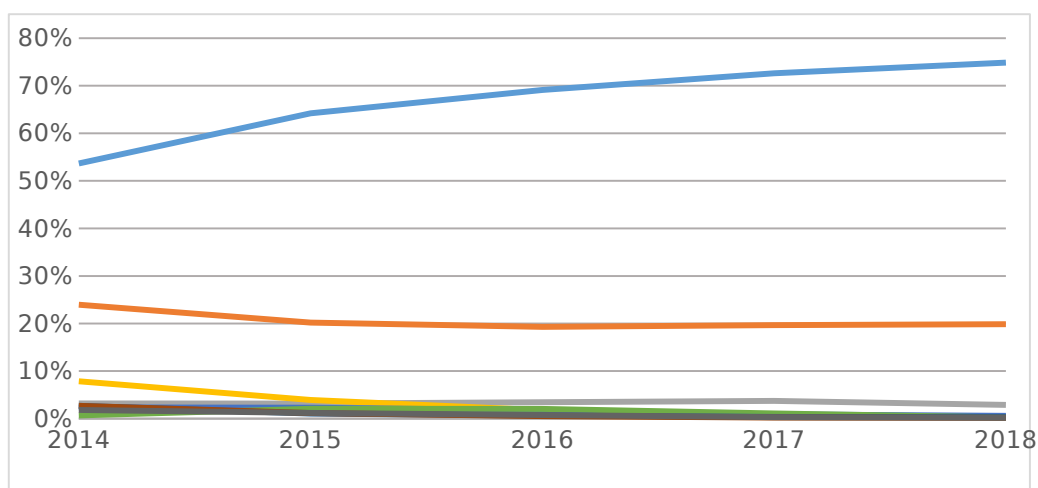


Рисунок 2.2. Рынок мобильных ОС (2014 г. – 2018 г.)

Помимо распространенности, Android обладает рядом существенных для разработчика преимуществ: возможность разработки на различных языках программирования, возможность вызова функций мобильных устройств, посредством использования стандартных вызовов API, множество вариантов для расширения функциональности приложения (включение дополнительной информации о географическом положении, контактных данных и т.д.), имеющиеся средства для удобной и легкой разработки.

1.5. Обоснование выбора СУБД

Общие сведения

Разрабатываемая система предполагает хранение и обработку данных, следовательно, необходимо средство для эффективного и удобного выполнения этих задач. Наилучшим образом для этого подходят базы данных.

Различают три основных типа баз данных: иерархическую, сетевую и реляционную. В настоящее время иерархические и сетевые модели данных уступают реляционной. Такое положение обосновано рядом имеющихся у реляционной модели преимуществ, среди которых: простота и наглядность в сравнении с другими двумя моделями, удобство физической реализации на ЭВМ, применение строгих правил при проектировании базирующихся на математическом аппарате, независимость данных.

Для взаимодействия с базами данных используются системы управления базами данных (СУБД). Для разработки базы данных было необходимо выбрать наиболее подходящую СУБД для поставленной задачи. В силу распространенности, для рассмотрения были выбраны: Firebird, MySQL, SQLite, PostgreSQL.

СУБД PostgreSQL

Свободно распространяемая СУБД, относится к объектно-реляционному типу. PostgreSQL отличается тем, что предоставляет объектно-ориентированный функционал, в том числе обладает поддержкой концепта ACID (атомарность, непротиворечивость, изолированность, сохранность данных).

Преимущества:

- Полная SQL-совместимость.
- Расширяемость: PostgreSQL — можно программно расширить за счёт хранимых процедур.
- Объектно-ориентированность: PostgreSQL — не только реляционная, но и объектно-ориентированная СУБД.

Недостатки:

- Производительность: в простых операциях чтения PostgreSQL может уступать своим соперникам, таким как MySQL.
- Популярность: из-за своей сложности инструмент не очень популярен.
- Подходит больше для работы со сложными структурами данных и не менее сложными процедурами.

СУБД SQLite

SQLite - встраиваемая компактная реляционная база данных.

Особенность SQLite в том, что она не использует парадигму клиент-сервер, поэтому её и называют «встраиваемой». Движок SQLite - не отдельно работающий процесс, а библиотека, с которой программа компонуется и движок становится составной частью программы.

Преимущества:

- Файловая: вся база данных хранится в одном файле, что облегчает перемещение.
- Стандартизированная: SQLite использует SQL.
- 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.

Недостатки:

- Отсутствие системы пользователей. Если необходимо обеспечить доступ к данным для нескольких пользователей, и к тому же различать их по правам доступа, то стоит выбрать другую СУБД.
- Отсутствие возможности увеличения производительности.
- Невозможность записи больших объемов данных - одним из ограничений SQLite являются операции записи.

СУБД FireBird

FireBird – свободно распространяемая СУБД. Поддерживает стандарты ANSI в синтаксисе языка SQL и позволяет работать на многих операционных системах.

Преимущества:

- Многоверсионная архитектура (параллельная обработка оперативных и аналитических запросов: читающие пользователи не блокируют пишущих);
- Соответствие требованиям ACID.

Недостатки:

- Отсутствие кеша итогов запросов, полнотекстовых индексов.
- Значительное падение производительности при росте внутренней фрагментации базы.

СУБД MySQL

Данная СУБД отличается большой скоростью, устойчивостью и легкостью использования. Предлагает широкий набор инструментов для разработки приложений. Благодаря своей популярности, существует много различных плагинов и расширений, которые могут упростить работу с системой. В силу того, что MySQL является серверной СУБД, приложения для доступа к данным, в отличие от SQLite работают со службами MySQL. Является идеальным решением для малых и средних приложений. Обычно MySQL используется в качестве сервера, к которому обращаются локальные или удалённые клиенты, однако в дистрибутив входит библиотека внутреннего сервера, позволяющая включать MySQL в автономные программы.

Преимущества:

- Простота: MySQL легко устанавливается. Существует много сторонних инструментов, облегчающих работу с БД.
- Широкий функционал: MySQL поддерживает большую часть функционала SQL.
- Безопасность: большое количество функций, обеспечивающих безопасность, которые поддерживаются по умолчанию.
- Масштабируемость: MySQL легко работает с большими объемами данных и легко масштабируется.
- Скорость: упрощение некоторых стандартов позволяет MySQL значительно увеличить производительность.
- Поддерживается неограниченное количество пользователей, одновременно работающих с базой данных, а количество строк в таблицах может достигать 50 млн.

Недостатки:

- Медленная разработка приложений в MySQL, о чем свидетельствуют негативные отзывы.
- Застой в разработке: хотя MySQL и является open-source продуктом, работа над ней сильно заторможена.

1.6. Обоснование выбора языков программирования

Для Windows приложения

C# - объектно-ориентированный язык программирования для платформы .NET. Основным постулатом C# является высказывание: "всякая сущность есть объект". Язык основан на строгой компонентной архитектуре и реализует передовые механизмы обеспечения безопасности кода.

C# относится к семье языков с C-подобным синтаксисом. Поддерживает черты объектно-ориентированного программирования. Является надежным и устойчивым языком, за счет использования «сборки мусора», безопасности типов и обработки исключений. C# разрабатывался "с нуля" и вобрал в себя много полезных свойств таких языков, как C++, Java, Visual Basic, а также Pascal, Delphi и др., избежав при этом многих отрицательных сторон своих предшественников. Мощная библиотека каркаса Framework.Net поддерживает удобство построения различных типов приложений и позволяет достаточно просто сохранять и получать информацию из баз данных.

Для мобильного приложения

Выбор языка программирования для разработки мобильного приложения во многом зависит от поставленных целей. Один из критериев для определения подходящего языка программирования является ОС под которую планируется разработка приложения. Для ОС Android одними из самых популярных языков являются: HTML5/JavaScript, Java, C/C++.

HTML5/JavaScript - являются сценарными языками программирования, адаптированными для разработки мобильных веб-приложений. Они несомненно обладают рядом преимуществ, например, кроссплатформенность, придание интерактивности веб-страницам. Поскольку разрабатываемое приложение является видом нативного приложения с тесным использованием функций ПО смартфона (камеры), данный стандарт не подходит для поставленной задачи.

C/C++ - нативные языки программирования, которые за счет использования NDK, также можно использовать для разработки под ОС Android. Дан-

ные языки, несомненно, не теряют своей актуальности и хорошо подходят для низкоуровневого программирования, позволяя создавать более производительные, быстрые, ресурсоемкие приложения. Код является более быстрым, за счет отсутствия необходимости использования виртуальной машины, но данное преимущество является существенным недостатком, сказывающимся в отсутствии переносимости. К тому же отсутствует объектно-ориентированный подход, что усложняет понимание и контроль кода.

Java - считается основным и официальным языком для программирования под ОС Android. Java является объектно-ориентированным, строго типизированным языком программирования, что дает ему преимущество перед другими средствами разработки. В связи с широким его применением, имеется множество руководств, учебников, инструкций, что может помочь при поиске необходимой информации или решения, при возникновении трудной ситуации. Java поддерживает множество API, что предоставляет широкий выбор инструментов для решений различных задач, включая актуальную для поставленной задачи использование возможностей камеры и обработки полученных изображений. К тому же Java используется в ряде мощных, современных интегрированных средств разработки (IDE), применение которых делает разработку более простой и контролируемой.

1.7. Обоснование выбора сред разработки

Данный выбор оказывает значительное влияние на удобство разработки, имеющийся инструментарий, возможности отладки и развертывания приложений на различных платформах, эффективность и качество реализуемых приложений.

Самым популярным решением для разработки под ОС Windows является среда Microsoft Visual Studio. Данная среда содержит компилятор языка программирования C#, предоставляет широкие возможности по отладке программ, включает в себя редактор исходного кода с поддержкой технологии IntelliSense, а также, что самое важное поддерживает технологию Windows

Forms, с помощью которой предполагается разработка графического интерфейса приложения.

В вопросе разработки мобильного приложения необходимо учесть, что разработка предполагается на языке Java для ОС Android. Самыми популярными являются следующие IDE: NetBeans, Eclipse, IntelliJ IDEA и, основанная на предыдущей, Android Studio.

Eclipse долгое время была основной средой для разработки приложений под Android. В связи с этим имеется множество ресурсов по вопросам Eclipse, видео, уроков, примеров, статей. Все это может оказать значительную помощь при разработке. Обладает стабильным, широким функционалом и развернутой системой плагинов. Однако имеется высокая вероятность столкнуться с ошибками несовместимости при установке этих плагинов. Несмотря на преимущества, по удобству данная IDE уступает не менее популярной IntelliJ IDEA.

IntelliJ IDEA – среда разработки, альтернативная Eclipse. Развитием занимается компания JetBrains. Компания постоянно занимается её расширением, обновлением с целью соответствовать самым последним потребностям рынка. Обладает большими возможностями и функционалом, удобством разработки, инструментами отладки и рефакторинга кода.

NetBeans – среда разработки, которая во многом не уступает другим IDE. Содержит много средств для редактирования, рефакторинга кода, включая «умное» автодополнение кода, присущее так же рассмотренным IntelliJ IDEA и Eclipse. Причем справляется с этой задачей несколько быстрее чем Eclipse, но медленнее чем, IntelliJ IDEA. Несмотря на всю свою мощь и перспективность NetBeans обладает существенным недостатком. Данная IDE является менее распространенной средой разработки, в связи с этим имеется меньше доступных источников информации.

Android Studio – развитие IntelliJ IDEA, разрабатываемая Google. Компания Google включила в Android Studio, помимо тех возможностей, которые предоставляет IntelliJ IDEA, дополнительные возможности (визуаль-

ные и функциональные), оказывающие поддержку при разработке. Встроенный SDK помогает оперативно избавляться от ошибок, связанных с некорректной версией API, причем студия сама предлагает нужную версию и её непосредственную установку.

Возможности интерфейса оказывают неизгладимую помощь при разработке. Здесь и возможности изменения отображения экрана, просмотр добавленных view элементов, отображение экрана на любом формате устройств.

Дизайн самой студии переработан, имеет ряд усовершенствований по сравнению с IntelliJ IDEA. Экраны логов, сообщений, свойств, занимают большую часть экрана и имеют возможность сворачивания в нижнюю панель, что очень удобно при разработке. Таким образом Android Studio лучше всего адаптирована для разработки под Android.

1.8. Описание реализуемой архитектуры клиент-сервер

Разрабатываемое приложение предполагает взаимодействие с удаленной базой данных согласно архитектуре клиент-сервер.

Архитектура клиент-сервер представляет собой распределенную структуру, основными элементами которой являются: сервер, выполняющий запросы и совокупность клиентов, инициализирующих эти самые запросы. Связь между сервером и клиентами при этом организуется с использованием определенного протокола. Сервер хранит ресурсы, необходимые клиентам.

В рамках разрабатываемой системы, сервер хранит базу данных с информацией о перечне контрольно-измерительных приборов, имеющих в подразделении предприятия. Сервер принимает, обрабатывает запросы от клиентов и отправляет ответ. Клиенты – мобильные устройства с приложением, формирующим запросы к серверу на получение информации о конкретном оборудовании для последующего отображения этой информации на экране.

На транспортно-сетевом уровне связь между серверами и клиентами осуществляется с использованием протокола TCP/IP. Поверх него, для обме-

на информацией, функционирует бинарный клиент-серверный протокол mysql.

1.9. Выбор средства для взаимодействия с базой данных

В Java одной из возможностей работы с удаленной базой данных является использование технологии JDBC. JDBC представляет собой Java API, то есть это совокупность классов и интерфейсов. Данная технология является унифицированной и позволяет одинаково хорошо работать с различными типами СУБД путем использования полиморфизма через реализуемые интерфейсы.

JDBC API делится на несколько уровней и включает в себя: прикладной уровень, уровень драйвера, описательный уровень. Конкретная реализация представляет собой JDBC драйвер для работы с конкретной СУБД. Данная архитектура представлена на рисунке 2.3.

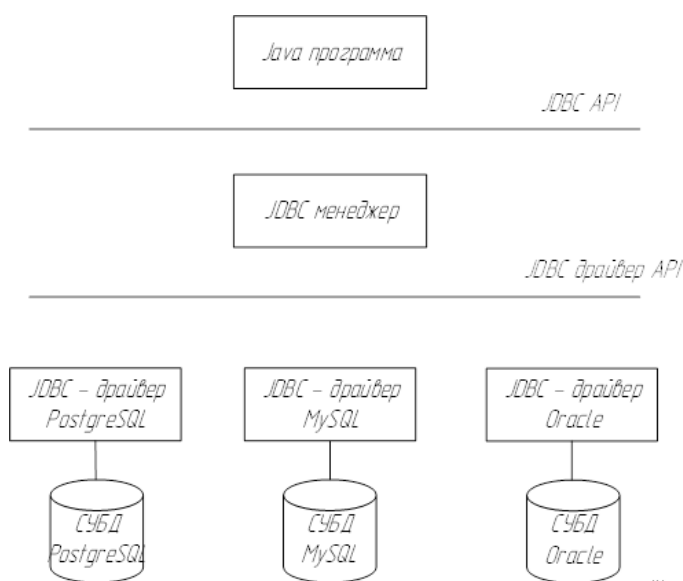


Рисунок 2.3. Архитектура JDBC

1.10. Информация о QR-коде

QR код представляет собой двумерный штрих-код с зашифрованной в нем информацией. Обычно штрих-код содержит информацию об объекте, к которому он непосредственно привязан. В рамках разрабатываемого проекта

было решено в качестве зашифрованной информации использовать заводской номер оборудования, поскольку он однозначно определяет устройство.

QR код состоит из четырех квадратов, расположенных к квадратной сетке на белом фоне. Код определяется камерой устройства как двумерное изображение. Три квадрата в углах изображения и меньшие синхронизирующие квадраты по всему коду позволяют нормализовать размер изображения, его ориентацию и угол. Точки подвергаются переводу в двоичные числа с проверкой по контрольной сумме.

Способ кодирования данных в QR коде осуществляется исходя из используемых символов. Различают следующие способы кодирования:

- Цифровое кодирование применяется, когда в качестве символов используются цифры от 0 до 9.
- Буквенно-цифровое кодирование применяется, когда кроме цифр необходимо зашифровать символы $\pm*/\%*.:,$ буквы латинского алфавита, пробел.
- Байтовое кодирование позволяет закодировать любые символы. Входной поток символов подвергается кодировке любого формата.

QR код имеет обязательные и не обязательные поля. Обязательные поля не содержат закодированной информации, они содержат информацию для декодирования. К списку обязательных полей относят:

- поисковые узоры;
- выравнивающие узоры;
- полосы синхронизации;
- уровень коррекции и код маски;
- код версии;
- обязательный отступ вокруг кода.

Оставшееся свободное место подвергается делению на столбцы, куда в последующем заносится информация. Информация заносится «змейкой». Первым делом заносится 1 бит информации в правый нижний квадрат, затем

в соседний левый, затем в тот, который был над правым и т.д. Занесение информации происходит снизу-вверх, а потом сверху вниз.

Выводы

Таким образом, исходя из требований к реализуемой системе, рассмотрения возможностей наиболее подходящих инструментов, вариантов разработки и последующего их сравнения было решено использовать следующие решения:

В качестве ОС для разработки были выбраны Windows и Android в силу имеющегося ряда преимуществ и учета конкретной целевой аудитории для которой она предназначена.

Исходя из требований к системе и характеристик каждой из рассмотренных СУБД в качестве оптимальной была выбрана MySQL.

В качестве инструментов для реализации серверной части было решено использовать: язык программирования C#, среду разработки Microsoft Visual Studio и технологию Windows Forms для реализации графического интерфейса.

Для разработки мобильного приложения было решено использовать следующие инструменты:

В качестве языка программирования был выбран язык программирования Java.

В качестве интегрированной среды разработки была выбрана адаптированная специально под ОС Android - Android Studio.

Для взаимодействия с удаленной базой данных было решено использовать технологию JDBC, поскольку данная технология является наиболее подходящей для решения поставленной задачи и к тому же проста и удобна в использовании.

2. ПРОЕКТИРОВАНИЕ КОМПОНЕНТОВ ПРОГРАММНОГО ПРОДУКТА

2.1. Разработка структуры системы

Разрабатываемая система состоит из следующих компонентов:

База данных – хранит всю необходимую информацию об имеющимся на предприятии оборудовании.

Серверная часть – представляет собой компьютер-сервер на базе ОС Windows с базой данных и приложением для добавления, изменения, получения, удаления информации об имеющимся на предприятии контрольно-измерительном оборудовании с возможностью вывода отчета в формате Excel.

Клиентская часть – мобильное приложение на базе ОС Android для получения информации об оборудовании из удаленной базой данных путем считывания его QR кода с возможностью генерации такого кода с последующим сохранением его в памяти телефона.

2.2. Разработка базы данных

База данных состоит из следующих таблиц:

- перечень оборудования;
- тип;
- тип-свойство;
- свойство;
- оборудование-свойство.

Структура базы данных представлена на рисунке 3.1.

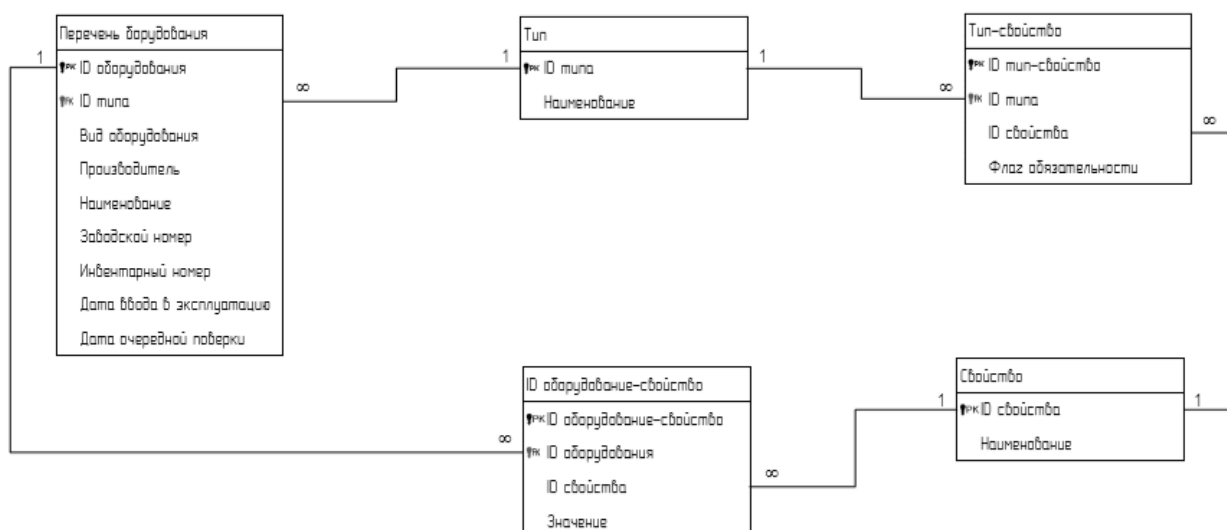


Рисунок 3.1. Структура базы данных

Ниже приведено описание каждой из таблиц.

Таблица 1 – «Перечень оборудования» - хранит общую информацию об оборудовании

Наименование поля	Тип данных	Признак ключа	Примечание
ID оборудования	int	PK	Уникальное значение
ID типа	int	FK	Уникальное значение. Указывает на запись в таблице Тип.
Вид оборудования	enum		Включает в себя: измерительное оборудование, индикаторный прибор, вспомогательное оборудование.
Производитель	varchar(100)		
Наименование	varchar(100)		
Заводской номер	varchar(50)		Служит в качестве зашифрованной информации в QR коде оборудования, поскольку однозначно определяет последнее.

Наименование поля	Тип данных	Признак ключа	Примечание
Инвентарный номер	varchar(50)		
Дата ввода в эксплуатацию	date		
Дата очередной поверки	date		

Таблица 2 – «Тип» - справочник для хранения всех типов оборудования

Наименование поля	Тип данных	Признак ключа	Примечание
ID типа	int	PK	Уникальное значение
Наименование	varchar(50)		Наименование типа

Таблица 3 – «Свойство» - справочник для хранения всех свойств оборудования

Наименование поля	Тип данных	Признак ключа	Примечание
ID свойства	int	PK	Уникальное значение
Наименование	varchar(50)		Наименование свойства

Таблица 4 – «Тип-свойство» - служит для обеспечения связи «многие ко многим» между таблицами «Тип» и «Свойство»

Наименование поля	Тип данных	Признак ключа	Примечание
ID тип-свойство	int	PK	Уникальное значение
ID типа	int	FK	Уникальное значение. Указывает на запись в таблице Тип.
ID свойства	int	FK	Уникальное значение. Указывает на запись в таблице Свойство.
Флаг обязательности	bit		Указывает на обязательность наличия свойства у данного типа оборудования

Таблица 5 – «Оборудование-свойство» - служит для обеспечения связи «многие ко многим» между таблицами «Перечень оборудования» и «Свойство»

Наименование поля	Тип данных	Признак ключа	Примечание
ID оборудование - свойство	int	PK	Уникальное значение
ID оборудования	int	FK	Уникальное значение. Указывает на запись в таблице «Перечень оборудования».
ID свойства	int	FK	Уникальное значение. Указывает на запись в таблице «Свойство».
Значение	varchar(500)		Содержит значение данного свойства.

2.3. Построение диаграммы вариантов использования Windows приложения

Данное приложение позволяет помимо получения информации об оборудовании, производить различные действия над ним. Подробное описание возможностей данного компонента системы удобно представить при помощи языка прецедентов. Прецедент представляет собой один из вариантов использования приложения. Совокупность же прецедентов помогает определить перечень действий для достижения необходимого результата. Разработанная диаграмма представлена на рисунке 3.2.

Use case диаграмма Windows приложения

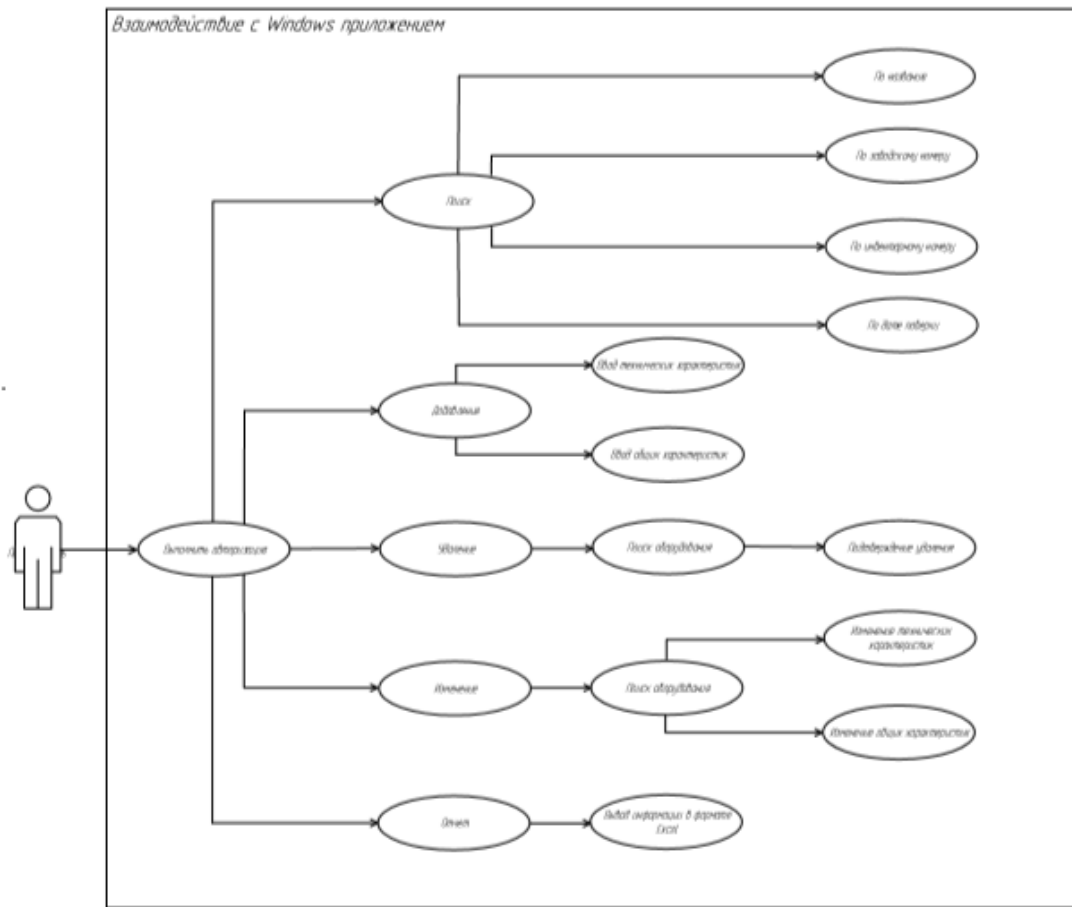
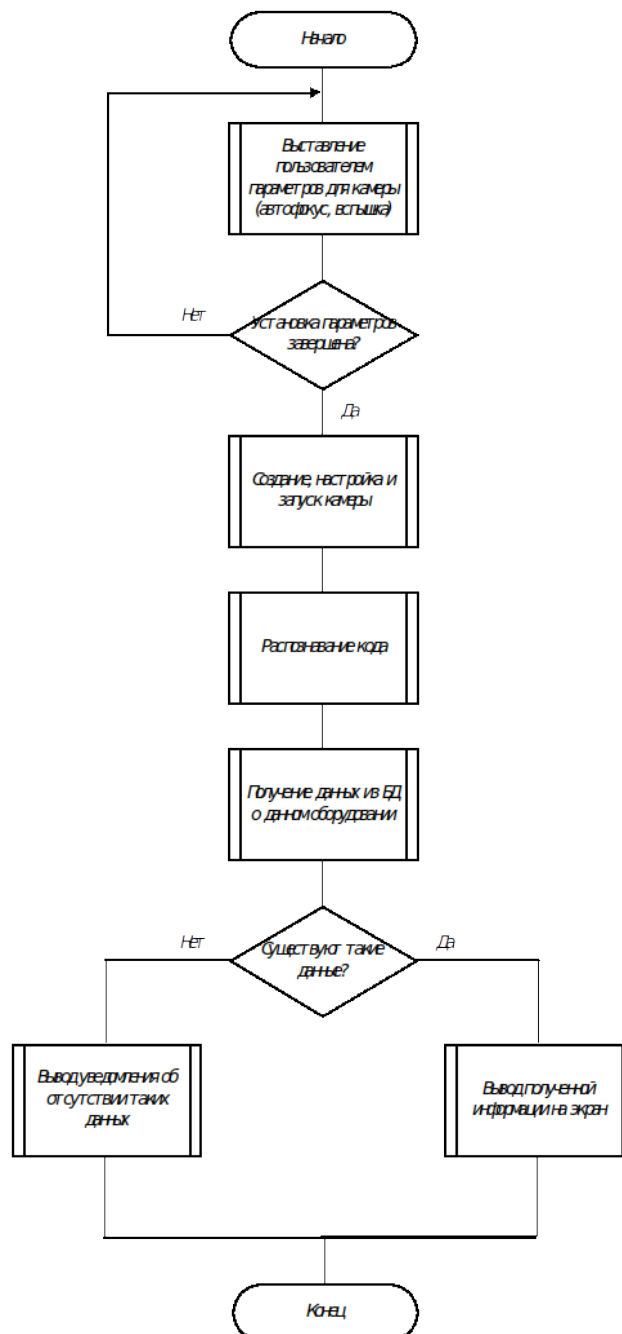


Рисунок 3.2. Use case диаграмма Windows приложения

2.4. Структура Android приложения

Мобильное приложение позволяет получать информацию о нужном оборудовании путем считывания его QR кода и при необходимости генерации изображения такого кода с последующим сохранением его в памяти телефона. Ниже приведена блок-схема иллюстрирующая алгоритм получения информации об оборудовании.



Блок-схема 1. Алгоритм получения информации об оборудовании

Данный компонент системы представляет собой Android приложение структура которого строится на совокупности описанных ниже модулей (активностей) и набора вспомогательных классов.

— Главная активность определена в файле MainActivity.java. Данная активность предназначена для вывода стартового экрана при запуске приложения, экрана авторизации пользователя. В случае успешного соединения происходит вызов следующей активности.

- Активность, описанная в файле `AssigningActivity.java` отвечает за окно выставления настроек для и вызов камеры с заданными параметрами.
- Активность, описанная в файле `InformationList.java` вызывается после считывания очередного QR-кода. Данная активность отвечает за получение информации из базы данных о приборе с соответствующим кодом и её выводе на экран мобильного устройства.
- Активность, описанная в файле `BarcodeCaptureActivity.java` вызывается после нажатия кнопки считывания QR-кода. Данная активность запускает превью камеры, выполняет обнаружение QR- кодов на данном превью, отображает их значения. При считывании кодов на каждый из них накладываются графические рамки, для определения позиции, идентификатора и размера. Все вышеперечисленные функции выполняются во вспомогательных классах, описанных далее.
- Вспомогательный класс `ConnectionDB.java` отвечает за подключение к удаленной базе данных.
- Класс `CameraSource.java` предоставляет функционирование камерой, для предварительного просмотра. Управляет заданными настройками, запускает камеру, обеспечивает поворот.
- Класс `CameraSourcePreview.java` отвечает за управление превью камеры. Её отображением, размерами, разрешением на запуск и т.д.
- Класс `GraphicOverlay.java` реализует отображение графических объектов поверх созданного превью. Поддерживает добавление, удаление, обновление элементов и управление масштабированием графического объекта относительно свойств превью камеры.
- Вспомогательный класс `BarcodeTrackerFactory.java` служит для реализации паттерна «Фабрика», для создания трекеров QR-кода. По одному трекеру на каждый код.
- Класс `BarcodeGraphicTracker.java` представляет трекер для обнаружения QR-кодов. Занимается отслеживанием кодов, наложением на них графики и удалением при необходимости.

— Класс BarcodeGraphic.java отвечает за отрисовку экземпляра накладываемого на код изображения с учетом его идентификатора, размера и положения.

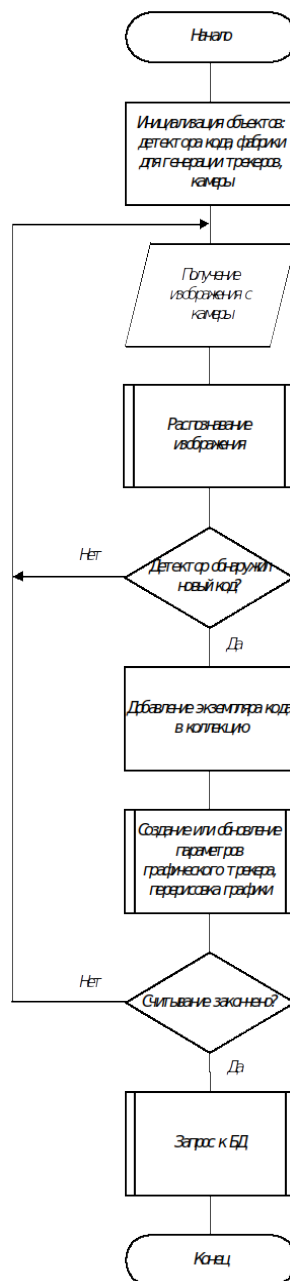
Mobile Vision API

Для распознавания QR-кодов в приложении используется стандартная библиотека Mobile Vision API. Таким образом нет необходимости интегрировать сторонний API в проект, все что необходимо это правильно интегрировать службы Google Play. Включает в себя два пакета: для распознавания лиц и штрих-кодов. Последняя используется в данной разработке. Данная API позволяет распознавать штрих-коды в любом положении экрана (ориентации) и в режиме реального времени. Поддерживает распознавание не только QR-кода, но и штрих-коды других 1D и 2D форматов (EAN-13, Code-39, ITF, PDF-417, AZTEC и т.д.).

Взаимодействие основных классов и интерфейсов

Процесс распознавания реализуется взаимодействием между следующими классами и интерфейсами: CameraSourcePreview, CameraSource, BarcodeDetector, MultiProcessor, BarcodeTrackerFactory, BarcodeGraphicTracker, BarcodeGraphic.

Ниже представлен алгоритм распознавания в виде блок-схемы.



Блок-схема 2. Алгоритм распознавания штрих-кода

На первом этапе, для отслеживания штрих-кода, создается детектор штрих кода – экземпляр BarcodeDetector. Используя созданный детектор, создается связанный с ним конвейер процесса, для дальнейшего получения результатов обнаружения. Мультипроцессор служит для создания трекера (BarcodeGraphicTracker <BarcodeGraphic>), для каждого штрих-кода, оказавшегося в зоне видимости камеры. BarcodeGraphic накладывает на изображение с камеры, ограничивающий прямоугольник и значение для каждого результата штрих-кода.

MultiProcessor – это компонент для работы с произвольным количеством обнаруженных элементов (штрих-кодами), т.е. позволяет одновременно сканировать и обнаруживать несколько штрих-кодов. По мере движения камеры, обновления для штрих-кодов направляются в каждый из соответствующих экземпляров BarcodeGraphicTracker, которые по средствам BarcodeGraphic осуществляют перерисовку конкретного штрих-кода. Когда штрих-код исчезает из зоны видимости, многопроцессор уничтожает связанный с ним экземпляр BarcodeGraphicTracker. Таким образом осуществляется динамическое создание / отслеживание / уничтожение индивидуального трекера для каждого кода.

Для отображения изображения с камеры, служит CameraSourcePreview. CameraSourcePreview отвечает за пользовательский интерфейс предварительного просмотра. CameraSource управляет получением изображений с CameraSourcePreview и передачей их детектору для последующей обработки.

Ниже приведена диаграмма описанного взаимодействия (рисунок 3.3).

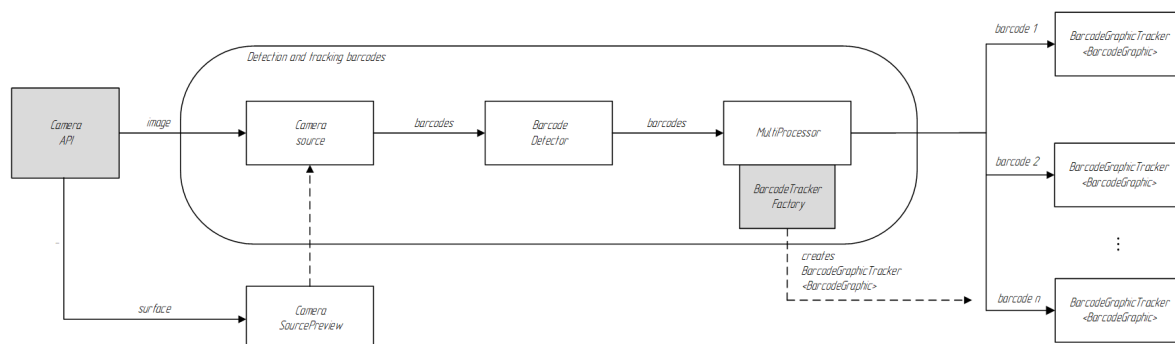
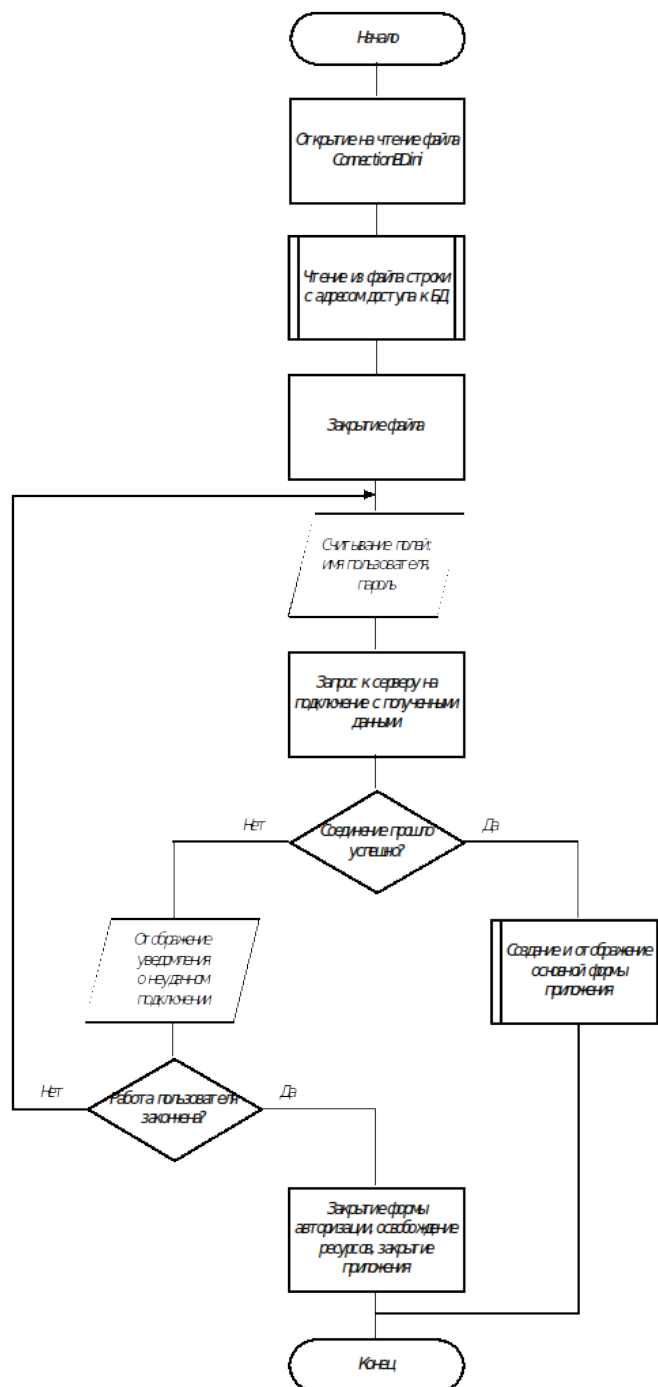


Рисунок 3.3. Диаграмма зависимости основных классов и интерфейсов

2.5. Организация сетевого взаимодействия.

В разрабатываемой системе клиенты - мобильные устройства с приложением, формирующим запросы к серверу на получение информации о конкретном оборудовании для последующего отображения этой самой информации на экране. Ниже приведена блок-схема иллюстрирующая алгоритм подключения клиента к серверу базы данных.



Блок-схема 3. Алгоритм подключения к серверу БД

На транспортно-сетевом уровне связь между серверами и клиентами осуществляется с использованием протокола TCP/IP. Поверх него, для обмена информацией, функционирует собственный бинарный клиент-серверный протокол mysql. Разграничение прав доступа реализуется средствами mysql.

Описанный стек протоколов представлен на рисунке 3.4.

<i>Прикладной</i>	<i>MySQL protocol</i>
<i>Транспортный</i>	<i>TCP</i>
<i>Сетевой</i>	<i>IP</i>
<i>Сетевых интерфейсов</i>	<i>Wi-Fi</i>

Рисунок 3.4. Стек протоколов разрабатываемой системы

Использование JDBC

Для организации сетевого взаимодействия сначала необходимо подключить драйвер JDBC для работы Android приложения с конкретным типом базы данных, в нашем случае с MySQL. Для организации такой работы необходимо:

- Произвести загрузку и регистрацию необходимого класса драйвера (jar библиотека).
- Указать путь к драйверу (classpath).

Основными интерфейсами и классами для работы через JDBC служат: Connection, Statement, ResultSet, DriverManager.

Класс DriverManager является посредником между программой и драйвером. Является основным интерфейсом JDBC. Основная функция – присоединить Java приложение к требуемому драйверу, для конкретной СУБД.

Connection служит для подключения к БД. Результатом успешного подключения служит созданный объект, описывающий данное соединение. Программа получает ссылку на этот объект для дальнейшей работы. Этот объект содержит характеристики состояние соединения с базой данных, и кроме того предоставляет средства для контроля транзакций.

Объекты классов, реализующих интерфейс Statement служат для создания и выполнения SQL запросов к БД. Служит контейнером для SQL выражений, включающих не только текст запроса, но и параметры выражения, его состояние.

Объекты классов, реализующих интерфейс `ResultSet` используются для получения результатов запросов, представляют собой коллекции строк.

Ниже приведена диаграмма взаимодействия между элементами JDBC, приложением и БД (рисунок 3.5).

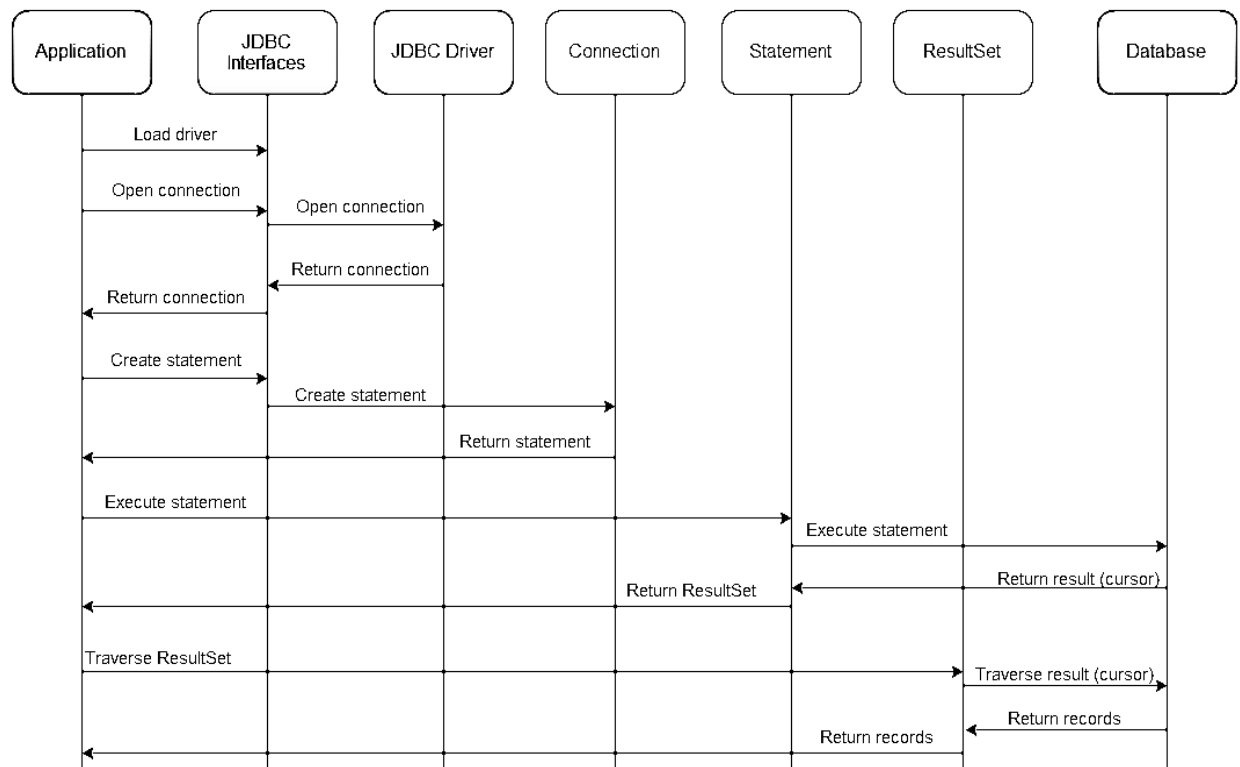


Рисунок 3.5. Взаимодействие между компонентами JDBC, приложением и БД.

Выводы

В результате были выполнены проектирование и реализация компонентов программного продукта: базы данных, серверной части и клиентской части. Полностью учтены функциональные требования технического задания.

3. ТЕСТИРОВАНИЕ И ИНТЕГРАЦИЯ КОМПОНЕНТОВ ПРОГРАММНОГО ПРОДУКТА

3.1. Тестирование системы

Тестирование системы производилось с использованием персонального компьютера в качестве сервера и мобильных устройств на базе ОС Android. На сервере тестировалась работа Windows приложения по учету оборудования и включало в себя проверку всех возможных вариантов использования с последующим выводом отчета. На мобильных устройствах производилось тестирование возможностей клиентского приложения, а также получение информации из базы данных с нескольких устройств одновременно.

Требования к Windows приложению

Для работы Windows приложения необходимо, чтобы компьютер обладал следующими минимальными характеристиками:

- процессор с тактовой частотой не менее 1 ГГц;
- ОЗУ 1 Гб;
- 2 Гбайт свободного пространства на диске;
- Microsoft Windows 7/8/8.1/10
- MySQL Server
- .NET Framework 4.5.2 или выше

Для взаимодействия с приложением необходимо наличие мыши или монитора.

Требования к мобильному приложению

Минимальные требования для работы мобильного приложения:

- мобильное устройство под управлением ОС Android версии 4.0 или выше;
- 9,6 Мб свободного пространства;
- наличие камеры;

Разрешения:

- просмотр сетевых подключений;

- доступ к камере;
- съемка фото;
- управление функциями вспышки и автофокуса.

Требования к программному обеспечению

Для работы приложения на сервере-компьютере с базой данных необходимо установить и настроить MySQL сервер.

3.2. Руководство администратора

Настройка доступа к базе данных

Для работы необходимо установить и настроить mysql сервер. Подробное руководство изложено в соответствующей документации <https://dev.mysql.com/doc/refman/8.0/en/windows-installation.html>. Для обеспечения доступа к удаленной базе данных сервер должен быть запущен. На уровне сервера необходимо определить уровень доступа пользователей к базе данных сервера. После добавления пользователей и ввода соответствующих разрешений, система безопасности MySQL будет самостоятельно осуществлять аутентификацию очередного пользователя при попытке доступа.

Для добавления пользователя и настройки для него определенных прав доступа необходимо выполнить следующие действия:

Добавить нового пользователя:

```
CREATE USER 'имя нового пользователя'@'адрес устройства'
IDENTIFIED BY 'пароль';
```

Для того, чтобы пользователь мог подключаться к базе данных с любого устройства, необходимо в качестве адреса указать символ %.

Установить привилегии для пользователя:

```
GRANT [тип прав] ON [имя базы данных].[имя таблицы] TO 'имя пользователя'@'адрес устройства';
```

Доступные привилегии:

- CREATE – Разрешить создание базы данных/таблиц
- SELECT – Разрешить производить выборку данных

- INSERT – Разрешить добавление новых записей в таблицы
- UPDATE – Разрешить изменение существующих записей в таблицах

- DELETE – Разрешить удалять записи из таблиц
- DROP – Разрешить удалять записи в базе данных/таблицах

Для того, чтоб изменения вошли в силу запустите команду обновления:
FLUSH PRIVILEGES;

3.3. Руководство пользователя

Руководство к Windows приложению

Для запуска приложения необходимо запустить на выполнение файл. «AccountingSystemCMequipment.exe». На экране появится окно авторизации (рисунок 4.1).

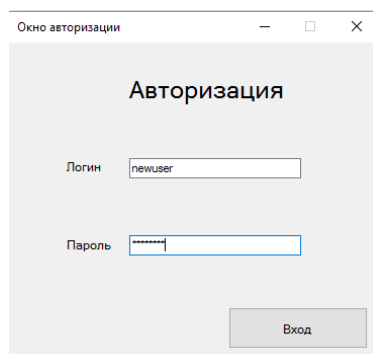


Рисунок 4.1. Окно авторизации

Если авторизация прошла успешно на экране появится окно главного меню (рисунок 4.2).

Главное меню

Поиск Добавление Изменение Удаление Отчет

Перечень оборудования

	Вид оборудования	Тип оборудования	Производитель	Наименование	Заводской номер	Инвентарный номер	Дата ввода в эксплуатацию	Дата очередного
▶	Измерительное...	Измерительный...	Семикон	ЦБ8535	D82234	7002733	10.02.2016	15.02
	Измерительное...	Анализатор	Rohde & Schwarz	R&S®FSW	HG2763	20039444	15.02.2016	16.02
	Измерительное...	Приемник	Rohde & Schwarz	R&S®ESCI	F928004	340400459	10.04.2016	24.04
	Измерительное...	Осциллограф	НПФ «Электро...	C1-65A	GH8273	720333	25.05.2016	
	Измерительное...	Генератор	Rohde & Schwarz	R&S®SMBV100	TY6333	30072345	01.01.2017	10.01
	Вспомогательн...	Антенна	Softlin	TL-ANT2408CL	HY633304	9200000042	28.02.2017	
	Вспомогательн...	Пробник	РМАП	РБ - 122/1	U827	3002723	02.03.2017	05.03
	Измерительное...	Токоосъемник	Rohde & Schwarz...	Токоосъемник E...	F45532PG	002738634	15.03.2017	
	Измерительное...	Мультиметр	Uni-Trend Group	UT118B	TWG6022	0983930002	28.03.2017	30.03
	Измерительное...	Генератор	НЗиФ	Г4-154	Y831019203	230005	24.04.2017	26.04
*								

< >

Рисунок 4.2. Главное меню

Кнопка «Поиск» позволяет осуществить поиск оборудования. При желании можно указать критерии поиска (рисунок 4.3).

Поиск

Укажите условия выбора из базы данных

Название оборудования
Анализатор

Заводской номер Инвентарный номер Дата проверки
 16.02.2016

Показать Отменить

Рисунок 4.3. Окно «Поиск»

Рисунок 4.4 иллюстрирует результат выполнения поиска оборудования.

Результат

Результат выполнения операции

	Вид оборудования	Тип оборудования	Производитель	Наименование	Заводской номер	Инвентарный номер	Дата эксп.
▶	Измерительное...	Анализатор	Rohde & Schwarz	R&S®FSW	HG2763	20039444	15.0
*							

Рисунок 4.4. Результат выполнения операции поиска

Для отображения технических характеристик, необходимо нажать в любом месте строки выбранного оборудования. Появится окно с имеющейся информацией о данном оборудовании (рисунок 4.5).

Анализатор спектра

Технические характеристики

Диапазон частот:

Погрешность измерения уровня:

Входная чувствительность при отстройке от несущей 10 кГц:

Опции:

RBW:

Масса:

Вход:

Подтвердить

Рисунок 4.5. Технические характеристики выбранного оборудования

Для обеспечения удобства и возможности сравнения можно производить поиск и открывать несколько окон с результатами одновременно.

Кнопка «Добавление» позволяет добавить оборудование в базу данных (рисунок 4.6).

Добавление нового оборудования

Выберите название оборудования

Подтвердить

Добавить

Вид оборудования

Производитель

Наименование

Заводской номер

Инвентарный номер

Дата ввода в эксплуатацию

Дата очередной поверки

Вернуться

Рисунок 4.6. Добавление нового оборудования

Сначала необходимо выбрать тип оборудования. Выбор определит окно, соответствующее данному типу для ввода технических характеристик (рисунок 4.7).

Осциллограф

Технические характеристики

Диапазон частот

20 Гц - 2 МГц

Опции

Каналы

X, Y, Z

Тип

универсальный

Масса

10 кг

Память

Частота дискретизации

100 МГц

Подтвердить

Рисунок 4.7. Окно ввода технических характеристик для осциллографа

После подтверждения введенных данных можно приступить к вводу общих характеристик (рисунок 4.8).

Рисунок 4.8. Ввод основных характеристик

В случае ввода некорректных данных появится значок, сообщающий об ошибке, рядом с соответствующим полем (рисунок 4.9).

Рисунок 4.9. Уведомление об ошибке

Кнопка «Добавить» будет не доступна до тех пор, пока не будут корректно заполнены все обязательные поля.

Кнопка «Изменение» позволяет изменить характеристики выбранного оборудования (рисунок 4.10).

Изменение

Условия выбора из базы данных

Название оборудования
Генератор

Заводской номер
Инвентарный номер
Дата поверки
10.01.2017

Применить

Вернуться

Наименование	Производитель	Заводской номер	Инвентарный номер	Дата ввода в эксплуатацию	Дата очередной поверки
ор	Rohde & Schwarz	R&S®SMBV100	TY6333	30072345	01.01.2017

Рисунок 4.10. Окно «Изменение» с указанием критериев

При желании можно указать критерии выбора оборудования из базы данных.

Для отображения характеристик, необходимо нажать в любом месте строки выбранного оборудования. Появится окно «Общие сведения» в котором можно вносить желаемые изменения (рисунок 4.11).

Общие сведения

Вид оборудования
Измерительное оборудование

Тип оборудования
Генератор

Производитель
Rohde & Schwarz

Наименование
R&S®SMBV100

Заводской номер
TY64555-32901

Инвентарный номер
30072345

Дата ввода в эксплуатацию
01.01.2017

Дата очередной поверки
10.01.2017

☒ Отобразить технические характеристики

Подтвердить

Рисунок 4.11. Окно изменения общих характеристик

Для изменения технических характеристик необходимо поставить галочку рядом с надписью «Отобразить технические характеристики» (рисунок 4.12).

Рисунок 4.12. Окно изменения технических характеристик для оборудования выбранного типа

Кнопка «Удаление» позволяет удалить выбранное оборудование.

Для выбора оборудования для удаления, необходимо нажать в любом месте строки выбранного оборудования. Появится окно «Подтверждения», в котором необходимо подтвердить или отклонить удаление (рисунок 4.13).

Рисунок 4.13. Окно «Подтверждение»

Кнопка «Отчет» позволяет вывести отчет, содержащий все таблицы базы данных в формате Excel (рисунок 4.14).

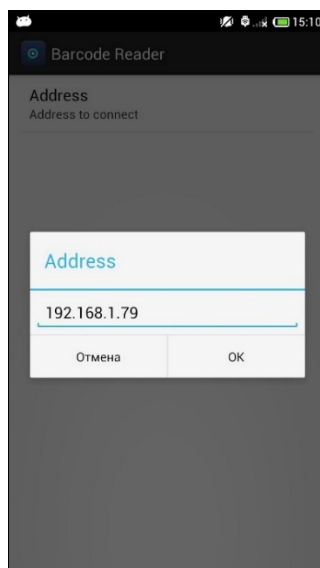


Рисунок 4.16. Ввод адреса подключения

В случае некорректного ввода данных для подключения или отсутствия последнего, внизу экрана появится уведомление о ошибке.

В случае успешного подключения, на экране отобразиться окно главного меню (рисунок 4.17).

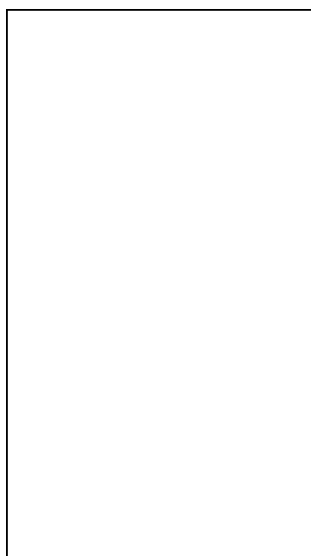


Рисунок 4.17. Окно главного меню

В случае выбора пункта сканирования кода, на экране отобразится окно для предварительного выставления настроек камеры. Имеется возможность включения использования автофокуса и/или вспышки (рисунок 4.18).

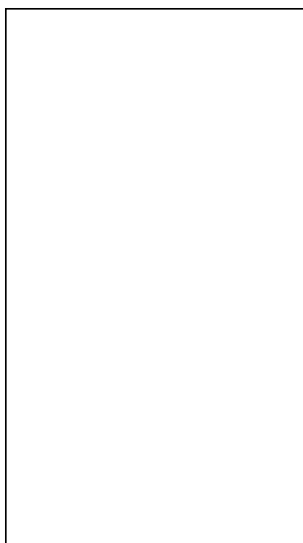


Рисунок 4.18. Окно указания настроек камеры

Нажатие клавиши «Read Barcode» активирует запуск камеры с указанными настройками для считывания QR-кода. Распознаваемые штрих-коды будут выделяться рамкой. Снизу рамки, области считывания, будут появляться данные о получаемой информации из кода (рисунок 4.19).

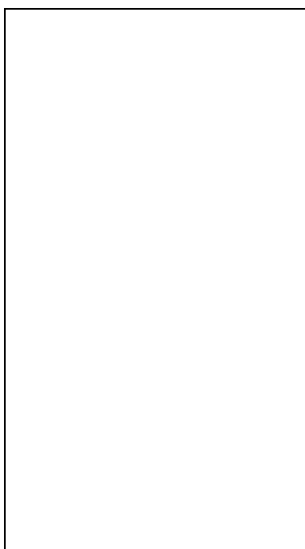


Рисунок 4.19. Процесс считывания QR-кода

Полученная информация об оборудовании отобразится на следующем экране (рисунок 4.20).

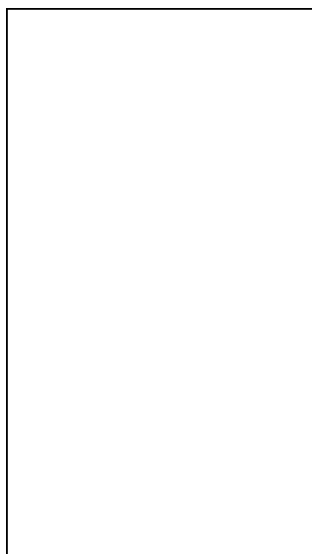


Рисунок 4.20. Полученная информация из базы данных

В случае, если на сервере базы данных не содержится информация о данном оборудовании, пользователь получит уведомление об отсутствии таких данных (рисунок 4.21).



Рисунок 4.21. Уведомление об отсутствии данных

При выборе в главном меню пункта генерации кода на экране появится соответствующее окно. В строке генерации необходимо ввести заводской номер оборудования и нажать клавишу «Generate». Сгенерированное изображе-

ние появится в центральной части экрана, а в нижней части экрана будет указан путь сохранения указанного изображения (рисунок 4.22).

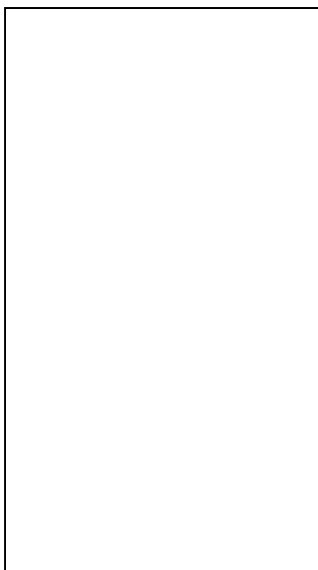


Рисунок 4.22. Окно генерации кода

Выводы

В результате были выполнены тестирование компонентов программного продукта. Разработана техническая документация.

ЗАКЛЮЧЕНИЕ

В результате выполнения выпускной квалификационной работы была создана система, позволяющая вести автоматизированный учет контрольно-измерительного оборудования в подразделении предприятия. Результат соответствует всем заранее определенным требованиям технического задания. Разработанная система обладает гибкостью, многофункциональностью и удобством в эксплуатации. Внедрение данной системы на предприятие позволит увеличить производительность труда за счет качественного анализа, планирования имеющихся ресурсов и уменьшения времени поиска необходимой информации, а также будет приносить экономическую выгоду в силу отсутствия затрат на материалы для производства бумажной документации.

Поскольку поставленная задача была выполнена в полном объеме дальнейшее её расширение не планируется. Однако разработанная архитектура позволяет при необходимости улучшить и расширить данную систему.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

Основная литература

1. Романенко, В.В. Объектно-ориентированное программирование : учебное пособие / В.В. Романенко ; Министерство образования и науки Российской Федерации, Томский Государственный Университет Систем Управления и Радиоэлектроники (ТУСУР). - Томск : Томский государственный университет систем управления и радиоэлектроники, 2014. - 475 с. : ил. - Библиогр.: с. 442. ; То же [Электронный ресурс]. - URL: <http://biblioclub.ru/index.php?page=book&id=480517>.
2. Николаев, Е.И. Объектно-ориентированное программирование : учебное пособие / Е.И. Николаев ; Министерство образования и науки Российской Федерации, Федеральное государственное автономное образовательное учреждение высшего профессионального образования «Северо-Кавказский федеральный университет». - Ставрополь : СКФУ, 2015. - 225 с. : ил. - Библиогр. в кн. ; То же [Электронный ресурс]. - URL: <http://biblioclub.ru/index.php?page=book&id=458133>.
3. Суханов, М.В. Основы Microsoft .NET Framework и языка программирования C# : учебное пособие / М.В. Суханов, И.В. Бачурин, И.С. Майоров ; Министерство образования и науки Российской Федерации, Федеральное государственное автономное образовательное учреждение высшего профессионального образования Северный (Арктический) федеральный университет им. М.В. Ломоносова. - Архангельск : ИД САФУ, 2014. - 97 с. : схем., табл., ил. - Библиогр. в кн. - ISBN 978-5-261-00934-4 ; То же [Электронный ресурс]. - URL: <http://biblioclub.ru/index.php?page=book&id=312313>.

Дополнительная литература

1. Мейер, Б. Объектно-ориентированное программирование и программная инженерия [Электронный ресурс] / Б. Мейер. — 3-е изд. — Электрон. текстовые данные. — М. : Интернет-Университет Информационных Техно-

логий (ИНТУИТ), Ай Пи Эр Медиа, 2019. — 285 с. — 978-5-4486-0513-0.

— Режим доступа: <http://www.iprbookshop.ru/79706.html>

2. **добавить свои источники, общим числом не менее 10**

3.