



Министерство науки и высшего образования Российской Федерации
Калужский филиал федерального государственного бюджетного
образовательного учреждения высшего образования
«Московский государственный технический университет
имени Н.Э. Баумана
(национальный исследовательский университет)»
(КФ МГТУ им. Н.Э. Баумана)

ФАКУЛЬТЕТ ИУК «ИНФОРМАТИКА И УПРАВЛЕНИЕ»

КАФЕДРА ИУК4 «ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ ЭВМ,
ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ»

РАСЧЕТНО-ПОЯСНИТЕЛЬНАЯ ЗАПИСКА К КУРСОВОЙ РАБОТЕ НА ТЕМУ:

Проектирование системы технологического процесса
создания тахографа

Студент группы ИУК4-62Б

(подпись, дата)

Е.В. Губин

(И.О. Фамилия)

Руководитель курсовой работы

(подпись, дата)

С.А. Глебов

(И.О. Фамилия)

Калуга, 2025

Министерство науки и высшего образования Российской Федерации
Калужский филиал федерального государственного бюджетного образовательного
учреждения высшего образования
«Московский государственный технический университет имени Н.Э. Баумана
(национальный исследовательский университет)»
(КФ МГТУ им. Н.Э. Баумана)

УТВЕРЖДАЮ
Заведующий кафедрой ИУК4
_____ (Ю.Е. Гагарин)
«__» _____ 2025г.

З А Д А Н И Е
на выполнение курсовой работы

по дисциплине **Проектирование систем хранения и обработки данных**

Студент группы **ИУК4-62Б Губин Егор Вячеславович**
(фамилия, имя, отчество)

Тема курсовой работы **Проектирование системы технологического процесса
создания тахографа**

Направленность КР **учебная**

Источник тематики **кафедра ИУК4**

Задание

Спроектировать структуры базы данных;

Разработать системное приложение;

Оформить графическую часть работы.

Оформление курсовой работы

Расчетно-пояснительная записка на 47 листах формата А4.

Перечень графического материала КР (плакаты, схемы, чертежи и т.п.):

– Структура базы данных – 1 лист формата А3;

– Визуальная часть приложения – 1 лист формата А3;

Дата выдачи задания «__» _____ 2025г.

Руководитель

(подпись, дата)

С.А. Глебов

(И.О. Фамилия)

Студент

(подпись, дата)

Е.В. Губин

(И.О. Фамилия)

Министерство науки и высшего образования Российской Федерации
Калужский филиал федерального государственного бюджетного образовательного
учреждения высшего образования
«Московский государственный технический университет имени Н.Э. Баумана
(национальный исследовательский университет)»
(КФ МГТУ им. Н.Э. Баумана)

КАЛЕНДАРНЫЙ ПЛАН на выполнение курсовой работы

по дисциплине Проектирование систем хранения и обработки данных

Студент группы ИУК4-62Б Губин Егор Вячеславович
(фамилия, имя, отчество)

Тема курсовой работы Проектирование системы технологического процесса
создания тахографа

№	Наименование этапов	Сроки выполнения этапов		Отметка о выполнении	
		план	факт	Руководитель	Куратор
1	Задание на выполнение	1-я нед.	1-я нед.		
2	Проектирование базы данных	4-я нед.	2-я нед.		
3	Разработка приложения и окончательное оформление графической части и расчетно-пояснительной записки	7-я нед.	6-я нед.		
4	Защита	11-я нед.	11-я нед.		

Студент _____ Руководитель _____
(подпись, дата) (подпись, дата)

СОДЕРЖАНИЕ

СОДЕРЖАНИЕ	4
ВВЕДЕНИЕ	5
1. МЕТОДЫ И ИНСТРУМЕНТЫ ПРОГРАММНОЙ ИНЖЕНЕРИИ	7
1.1. Техническое задание	7
1.2. Анализ существующих аналогов	11
1.3. Перечень задач, подлежащих решению в процессе разработки	12
1.4. Обоснование выбора языков программирования	13
1.5. Обоснование выбора сред разработки	15
1.6. Описание реализуемой архитектуры	17
Выводы	17
2. ПРОЕКТИРОВАНИЕ КОМПОНЕНТОВ ПРОГРАММНОГО ПРОДУКТА	19
2.1. Разработка структуры системы	19
2.2. Разработка базы данных	20
2.3. Структура десктоп-приложения	31
Выводы	33
3. ТЕСТИРОВАНИЕ И ИНТЕГРАЦИЯ КОМПОНЕНТОВ ПРОГРАММНОГО ПРОДУКТА	35
3.1. Тестирование системы	35
3.2. Руководство администратора	36
3.3. Руководство пользователя	41
Выводы	43
ЗАКЛЮЧЕНИЕ	44
СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ	45
Основная литература	45
Дополнительная литература	45

ВВЕДЕНИЕ

Тахограф — это прибор, фиксирующий скорость, время движения, остановки и другие параметры, обеспечивая тем самым прозрачность и контроль за деятельностью водителя. В связи с этим возрастают требования к качеству, надёжности и технологичности производства тахографов, что делает актуальным вопрос проектирования эффективного технологического процесса их создания.

Актуальность темы определяется необходимостью оптимизации производства тахографов на предприятиях с целью повышения производительности, качества выпускаемой продукции и соответствия нормативным требованиям. Внедрение современных методов проектирования технологических процессов позволяет сократить издержки, улучшить структуру производственного цикла и повысить конкурентоспособность предприятия.

Объектом работы является процесс производства тахографа на предприятии.

Предметом работы выступает система технологического процесса создания тахографа, включающая последовательность операций, методы контроля качества и логистику производства.

Целью работы является разработка ПО для системы технологического процесса создания тахографа, обеспечивающую эффективное и качественное производство устройства с учетом требований к точности и надёжности.

Для достижения поставленной цели необходимо решить следующие задачи:

1. Проанализировать требования к организации рабочих мест в технологическом процессе производства тахографа.
2. Определить функции и роль программного обеспечения в обеспечении эффективности и контроля технологического процесса.
3. Разработать архитектуру программного обеспечения для автоматизации рабочих мест на различных этапах производства тахографа.

4. Реализовать модули ПО для сбора, обработки и отображения технологических данных (например, контроль сборки, регистрация тестов, логирование операций и др.).
5. Обеспечить взаимодействие ПО с оборудованием и базами данных, используемыми на производственном участке.
6. Разработать интерфейс пользователя с учётом специфики работы операторов и инженеров на производстве.
7. Провести тестирование и отладку программного обеспечения на модельной или реальной части технологического процесса.
8. Оценить эффективность внедрения разработанного ПО и его влияние на производительность и качество изготовления тахографов.

1. МЕТОДЫ И ИНСТРУМЕНТЫ ПРОГРАММНОЙ ИНЖЕНЕРИИ

1.1. Техническое задание

Наименование системы

Настоящее Техническое задание определяет требования и порядок создания «Системы технологического процесса сборки тахографа».

Основания для разработки

Сборка тахографа осуществляется в несколько этапов. Каждый этап производится на отдельном рабочем месте и представляет из себя установку компонента на тахограф. На предприятии предусмотрено несколько исполнений (комплектаций) тахографа – в зависимости от исполнения на рабочих местах устанавливаются определённые компоненты. Производимая комплектация определяется сменным планом, который хранится на сервере предприятия. Данные рабочие места являются этапами сборки и отличаются только набором компонентов для данного рабочего места. На конкретном рабочем месте должен работать оператор, которого научили операции на этом рабочем месте, значит необходима реализация прав для каждого рабочего места. При производстве продукт может быть бракованным из-за некомпетентности или невнимательности оператора – необходима прослеживаемость технологического процесса. Для масштабирования или корректировки технологического процесса необходим конструктор создания (корректировки) рабочего места.

Исполнитель

Исполнителем проекта является студент Калужского филиала МГТУ им. Н. Э. Баумана, факультета ИУК, группы ИУК4-62Б, Губин Егор Вячеславович.

Краткая характеристика области применения

Разрабатываемая система предназначена для применения в области производства. Её цель – создание универсального ПО для этапов сборки тахографа и ПО, являющегося конструктором технологического процесса.

Целевая аудитория

Система разделения платежей ориентирована на рабочих в цехе и инженеров-технологов, которые управляют производством. Конечной целью является собранный продукт или всевозможный брак.

Назначение системы

Разрабатываемая система обеспечивает автоматическую прослеживаемость производства, конфигурирование рабочих мест, установка компонентов и прозрачность производства.

Цели создания системы

Целью разработки системы является получение конечного продукта, отслеживание производства и управление производством.

Плановые сроки начала и окончания работы по созданию системы

Планируемые сроки начала и окончания работы над работой:
_____.

Требования к надежности

Надежность программного продукта для веб-приложения является критически важной для обеспечения устойчивого функционирования и защиты данных пользователей.

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 системы должны соответствовать стандартам, предъявляемым к техническим средствам в сфере развлекательной индустрии и веб-технологий.

Требования к квалификации и численности персонала

Система требует наличия пользователей двух категорий – рядовых пользователей, которых необходимо обучить операции на рабочем месте и администраторов (они же инженеры технологи), необходимые для конфигурирования технологического процесса и обучения нового персонала.

Требования к защите информации и программ

Защита информации осуществляется разграничением прав доступа.

Предусмотрены следующие права:

- 0 – не имеет права работать на этом рабочем месте
- 1 – имеет право работать на этом рабочем месте
- 2 – имеет права администратора на этом рабочем месте

- 3 – общий уровень прав, самый главный администратор, имеет право работать на любом рабочем месте, создавать план и конфигурировать технологический процесс
- 255 – больше не работает на предприятии

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;

2. Графическая часть - 2 листа формата А3 включающие в себя:

- структурные схемы;
- основные алгоритмы;

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 документации 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, сопутствующего ПО для разработки, 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

На текущий момент на предприятии есть схожий технологический процесс по созданию электроусилителя для автомобиля Lada Granta. Его структура такая же, как и в нашей системе. Оба процесса имеют одинаковые организационно-технические этапы (КФ, ПСИ, Предпродажная подготовка, прошивка), помимо этапов сборки, так как это принципиально разные изделия. Ключевым отличием является прослеживаемость производства: в техпроцессе сборки электроусилителя отсутствует прослеживаемость брака, компонентов и операторов. Так же отсутствует конструктор, обеспечивающий масштабирование и изменение процесса. Таким образом, наш техпроцесс, в

отличие от техпроцесса создания ЭУР, обеспечивает гибкость, прозрачность и масштабируемость производства.

1.3. Перечень задач, подлежащих решению в процессе разработки

При производстве необходимо «электронная подпись» оператора на совершении конкретной операции над изделием, чтобы в случае брака отследить, на каком этапе появился брак и кто его допустил. Так же «электронная подпись» позволяет отследить, сколько изделий прошло через оператора рабочего места за смену или за какой-нибудь другой промежуток времени. Для решения прослеживаемости операторов реализована идея идентификации оператора через его рабочий пропуск, который имеет уникальный ID на предприятии. При работе на рабочем месте пропуск оператора лежит на специальном считывателе карт, который при совершении операции считывает ID карты и тем самым подписывает оператора в базе данных на данное изделие.

Для задачи конфигурирования технологического процесса создано специальное ПО, на котором может работать только организатор техпроцесса (инженер-технолог) с наивысшим уровнем прав. Это ПО обеспечивает:

- Создание рабочего места
- Конфигурирование рабочего места, а именно изменение используемых компонентов, изменение их количества для операции
- Создание плана на смену и приоритетного плана
- Управление имеющимися компонентами (к какому исполнению принадлежит, сколько компонентов, постоянный или переменный)

Для отслеживания компонентов, их количества и их партии реализована идея сканирования специальных этикеток на лотке с компонентами, которые приклеиваются заранее. Этикетка представляет из себя баркод с номером партии компонентов и сканируется сканером Saotron 50. Эти данные так же пишутся в базу данных.

Так как рабочие места сборки выполняют одни и те же функции, было принято решение реализации лишь одного проекта, который конфигурируется с помощью специального файла, который как раз и определяет рабочее место, операцию, список компонентов и связи прав и операторов для него. Этот файл помещён в защищённый паролем архив, чтобы оператор не смог произвести в нём изменения.

В целом для прозрачности производства почти каждое действие оператора, состояние ПО, операции на рабочих местах, учет компонентов и брака фиксируется в базе данных, находящейся на виртуальном сервере предприятия.

1.4. Обоснование выбора языков программирования

В рамках разработки программного обеспечения для автоматизации рабочих мест технологического процесса создания тахографа был произведён выбор языков программирования и сопутствующих технологий, отвечающих требованиям надёжности, удобства реализации, совместимости с используемым оборудованием и простоты сопровождения. Основными используемыми средствами стали язык программирования C# с использованием Windows Forms (WinForms) для создания графического интерфейса, а также СУБД MySQL для хранения и управления производственными данными.

Язык C# представляет собой современный объектно-ориентированный язык программирования, активно используемый в разработке настольных приложений под операционные системы Windows. Он обладает высоким уровнем интеграции с платформой .NET Framework, что делает его удобным и надёжным средством при создании корпоративного программного обеспечения.

Библиотека WinForms была выбрана как наиболее подходящая технология для быстрой разработки настольных приложений с графическим интерфейсом. Она предоставляет широкие возможности по созданию интерфейсов пользователя с элементами управления, необходимыми для производственных

приложений: таблицы, поля ввода, кнопки, меню, вкладки и т. д. Основными преимуществами использования WinForms являются:

- простота и высокая скорость разработки;
- надёжность и стабильность при использовании в промышленной среде;
- возможность запуска на большинстве рабочих станций под управлением Windows без необходимости установки дополнительных зависимостей.

Программное обеспечение предполагает интеграцию с внешними устройствами, такими как сканеры штрихкодов и считыватели смарт-карт, подключаемыми через последовательный порт (COM). Для взаимодействия с такими устройствами в C# используется класс `SerialPort`, входящий в стандартную библиотеку `.NET`. Он обеспечивает:

- прямой доступ к COM-портам;
- обработку событий получения данных;
- настройку параметров порта (скорость передачи, контроль чётности и т. д.).

Это позволяет без дополнительных библиотек обеспечить приём и обработку данных от устройств, используемых в производственном процессе (например, при идентификации сотрудника или отслеживании деталей по штрихкоду).

Для хранения информации о деталях, этапах сборки, операторах, результатах тестирования и другой технологической информации используется система управления базами данных MySQL. Она выбрана по следующим причинам:

- высокая производительность и надёжность при работе с большими объёмами данных;
- бесплатное распространение и открытый исходный код;
- широкая поддержка и документация;
- совместимость с C# через официальные коннекторы и сторонние ORM-библиотеки (например, `MySql.Data`).

Использование MySQL позволяет централизованно хранить и обрабатывать производственные данные, обеспечивая их доступность как для операторов, так и для администраторов системы. Взаимодействие между приложением на C# и базой данных реализуется через библиотеку MySql.Data, которая предоставляет удобные средства выполнения SQL-запросов, работы с транзакциями и обработки ошибок.

1.5. Обоснование выбора сред разработки

Для реализации программного обеспечения, предназначенного для автоматизации рабочих мест в технологическом процессе производства тахографов, были выбраны современные и проверенные средства разработки. Основными критериями при выборе программных средств стали: удобство разработки, совместимость с выбранными технологиями, наличие инструментов отладки и диагностики, а также широкая поддержка со стороны сообщества и документации. В процессе разработки использовались следующие программные среды:

В качестве основной среды разработки для написания клиентского приложения на языке C# с использованием WinForms была выбрана Visual Studio 2019 — одно из самых мощных и функциональных решений для создания программ под платформу .NET.

Основные преимущества Visual Studio 2019:

- Полная интеграция с .NET Framework и библиотекой Windows Forms.
- Удобная система визуального проектирования интерфейса.
- Инструменты для отладки, профилирования и тестирования приложений.
- Поддержка автоматической генерации кода, IntelliSense и рефакторинга.
- Возможность быстрого подключения и настройки сторонних библиотек, таких как MySql.Data для работы с MySQL.

Visual Studio 2019 обеспечивает надёжную и комфортную среду для разработки приложений промышленного уровня с поддержкой всех необходимых инструментов для реализации, отладки и сопровождения.

Для работы с базой данных MySQL использовалась официальная среда MySQL Workbench, предоставляемая разработчиками MySQL. Эта среда обеспечивает удобный графический интерфейс для проектирования, администрирования и сопровождения баз данных.

Преимущества MySQL Workbench:

- Визуальное моделирование структуры базы данных (ER-диаграммы).
- Возможность написания и тестирования SQL-запросов.
- Мониторинг состояния сервера и управление пользователями.
- Импорт и экспорт данных в различных форматах.

MySQL Workbench позволила эффективно разрабатывать структуру БД, управлять таблицами, индексами, представлениями и выполнять тестирование запросов ещё до интеграции с клиентским приложением.

Для отладки и настройки взаимодействия с внешними устройствами, подключаемыми через COM-порты (такими как сканер штрихкодов и считыватель карт), использовалась специализированная утилита AccessPort.

Эта программа предоставляет функциональность для:

- Открытия и мониторинга последовательных портов.
- Отправки и приёма данных в ручном или автоматическом режиме.
- Анализа обмена данными с оборудованием (включая ASCII, HEX и бинарные форматы).
- Настройки параметров порта (скорость передачи, биты данных, чётность, стоп-биты).

AccessPort использовалась на этапе тестирования взаимодействия с оборудованием и позволила выявить и корректно настроить параметры подключения перед их интеграцией в приложение через класс SerialPort в C#.

1.6. Описание реализуемой архитектуры

Разработка программного обеспечения для рабочих мест в технологическом процессе создания тахографа требует чёткого структурирования задач и функциональности. Для этого была спроектирована модульная архитектура, основанная на распределении технологического процесса по этапам и разбиении общей задачи автоматизации на более мелкие, функционально завершённые компоненты.

Реализуемая архитектура основывается на последовательной модели производственного цикла, где каждый этап технологического процесса сопровождается определёнными действиями оператора и поддерживается соответствующим программным модулем. Таким образом, каждая стадия сборки тахографа отображается в ПО как логически обособленный блок, обеспечивающий контроль, учёт и фиксацию результатов работ.

Основные принципы архитектурного решения:

- Модульность — каждая часть процесса реализована в виде отдельного программного модуля (форма, окно, компонент).
- Последовательность выполнения — обеспечивается логика перехода от одного этапа к другому, исключающая ошибки последовательности действий.
- Унификация взаимодействия — взаимодействие модулей с базой данных и внешними устройствами реализовано по единым шаблонам.
- Интерфейсная простота — пользовательский интерфейс ориентирован на минимизацию ошибок при работе операторов и упрощённый ввод данных.

Выводы

Таким образом, исходя из требований к разрабатываемому программному обеспечению для автоматизации технологического процесса сборки тахографа, была сформулирована архитектура системы и определены основные средства разработки. Проект предполагает создание программного комплекса, который

обеспечит автоматизацию и контроль всех этапов сборки, включая регистрацию операций, управление правами доступа и фиксацию брака.

Архитектура системы основана на клиент-серверной модели с использованием базы данных MySQL для централизованного хранения данных. Клиентская часть будет реализована на платформе WinForms с использованием языка C#, что позволяет создать удобный пользовательский интерфейс и эффективно взаимодействовать с оборудованием через COM-порты. Важно отметить, что выбранный стек технологий обеспечит гибкость и возможность расширения системы, что будет полезно для внедрения новых функций и интеграции с другими производственными системами в будущем.

Для разработки выбранная среда Visual Studio 2019, которая является мощным инструментом для создания приложений на C# и работы с базами данных. Она предоставляет удобные инструменты для разработки, отладки и тестирования программного обеспечения, что ускоряет процесс разработки и повышает его качество.

Архитектура и выбранные технологии обеспечивают реализацию всех необходимых функциональных требований и гарантируют высокую производительность и масштабируемость системы.

2. ПРОЕКТИРОВАНИЕ КОМПОНЕНТОВ ПРОГРАММНОГО ПРОДУКТА

2.1. Разработка структуры системы

Система представляет из себя ПО, которое регистрирует почти любое действие оператора на рабочем месте. Подпись оператора осуществляется через рабочий пропуск, который прикладывается к считывателю карт. При наличии пропуска на считывателе карт и необходимых прав оператора начинается работа на этом месте.

Ключевым компонентом является сканер баркодов. С его помощью фиксируется партия компонентов (штрих-код на лотке компонентов) и записывается в базу данных. Одновременно с информацией о партии записывается и номер карты оператора.

Как только информация штрих-кода записалась, необходимо сделать операцию успешной или неуспешной. В случае успешной операции можно подтвердить, просканировав ещё раз штрих-код на лотке или нажать на кнопку. В случае неуспешной операции подразумевается брак. Он бывает двух типов: брак производства – брак, допущенный на предыдущих местах сборки или после операции на текущем рабочем месте (при этом варианте можно забраковать компоненты текущие или используемые на предыдущих рабочих местах); брак поставщика – брак, обнаруженный до начала операции на данном рабочем месте (перечень компонентов для текущего рабочего места).

В случае обнаружения отсутствия карты начинается перерыв и рабочая область формы сворачивается во избежание несанкционированного доступа.

Конструктор с точки зрения карт устроен аналогично, только здесь проверяется иной уровень прав доступа, самый высокий. Оно представляет из себя отражение каждого рабочего места или всех компонентов и исполнений в таблицах. Эти таблицы можно полностью менять. Можно полностью составить технологический процесс сборки с самого начала.

База данных является централизованной и располагается на виртуальном сервере предприятия.

2.2. Разработка базы данных

Ввиду того, что база данных представляет из себя централизованную систему для всех рабочих мест, необходимо организовать поочерёдный доступ к ней. Для этого устанавливается самый строгий уровень транзакций – Serializable. При этом уровне транзакций клиенты обслуживаются по очереди, без возможности любого доступа к базе.

Есть множество мест сборки изделия, поэтому места сборки нумеруют по порядку в формате «tf_<номер места>». Формат таблиц для каждого такого места сборки одинаковый, отличается лишь номером.

База данных состоит из следующих таблиц:

- seats (места)
- persons (операторы)
- access_rights (права доступа)
- seats_persons (операторы и места)
- part_numbers (исполнения)
- plans (планы)
- seats_remains (остатки на местах)
- component_names (имена компонентов)
- components (компоненты)
- components_seats (компоненты на местах)
- components_variations (компоненты и вариации)
- defectivity_types (типы брака)
- defectivity (брак)
- tf_1 (таблица операций рабочего места)
- logs_types (типы логов)
- logs (логи)

Ниже приведена структура базы данных.

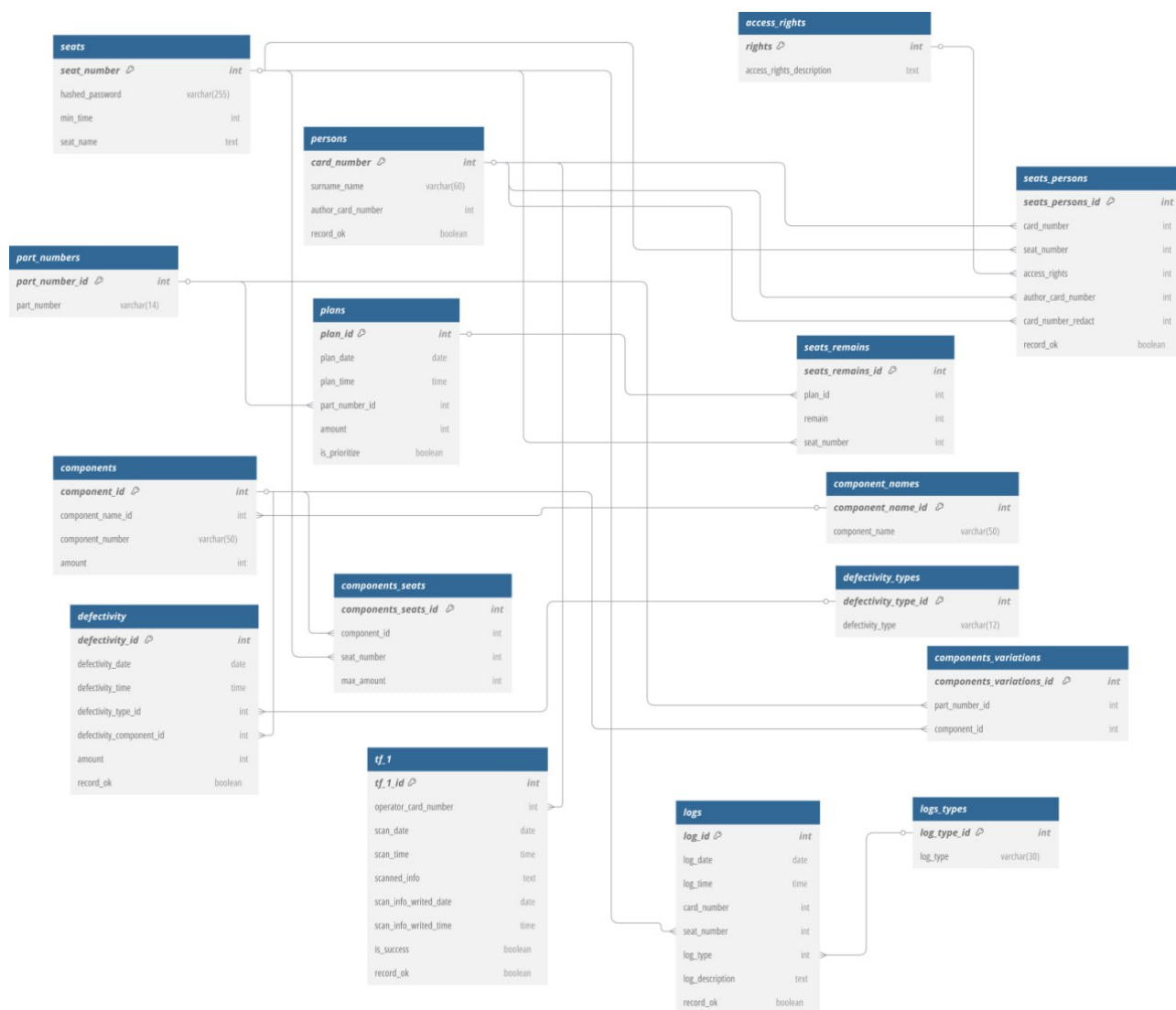


Рисунок 1.1 Структура базы данных

Ниже приведено описание каждой из таблиц.

Таблица 1 – «seats» – хранит общую информацию о рабочих местах

Наименование поля	Тип данных	Признак ключа	Примечание
seat_number	int unsigned	PK	Уникальное значение
hashed_password	varchar(255)		Не может быть NULL
min_time	int unsigned		Не может быть NULL
seat_name	text		Не может быть NULL

Таблица 2 – «persons» – хранит общую информацию об операторах

Наименование поля	Тип данных	Признак ключа	Примечание
card_number	int unsigned	PK	Уникальное значение
surname_name	varchar(60)		Не может быть NULL
author_card_number	int unsigned		Не может быть NULL
record_ok	boolean		

Таблица 3 – «access_rights» – хранит общую информацию об уровнях прав

Наименование поля	Тип данных	Признак ключа	Примечание
rights	int unsigned	PK	Уникальное значение
access_rights_description	text		Не может быть NULL

Таблица 4 – «seats_persons» – хранит информацию о рабочих местах, операторах и прав этих операторов на этих рабочих местах

Наименование поля	Тип данных	Признак ключа	Примечание
seats_persons_id	int unsigned	PK	Уникальное значение
card_number	int unsigned	FK	Не может быть NULL, ссылается на поле card_number таблицы persons, при удалении записи из таблицы persons соответствующая запись из таблицы seats_persons также будет удалена

seat_number	int unsigned	FK	Не может быть NULL, ссылается на поле seat_number таблицы seats, при удалении записи из таблицы seats соответствующая запись из таблицы seats_persons также будет удалена
access_rights	int unsigned	FK	Не может быть NULL, ссылается на поле rights таблицы access_rights, при удалении записи из таблицы access_rights соответствующая запись из таблицы seats_persons также будет удалена
author_card_number	int unsigned	FK	Не может быть NULL, ссылается на поле card_number таблицы persons, при удалении записи из таблицы persons соответствующая запись из таблицы seats_persons также будет удалена
card_number_redact	int unsigned	FK	Не может быть NULL, ссылается на поле card_number

			таблицы persons, при удалении записи из таблицы persons соответствующая запись из таблицы seats_persons также будет удалена
record_ok	boolean		

Таблица 5 – «part_numbers» – хранит все исполнения для изделия

Наименование поля	Тип данных	Признак ключа	Примечание
part_number_id	int unsigned	PK	Уникальное значение
part_number	varchar(14)		Не может быть NULL

Таблица 6 – «plans» – хранит текущие и прошедшие планы

Наименование поля	Тип данных	Признак ключа	Примечание
plan_id	int unsigned	PK	Уникальное значение
plan_date	date		Не может быть NULL
plan_time	time		Не может быть NULL
part_number_id	int unsigned	FK	Не может быть NULL, ссылается на поле part_number_id таблицы part_numbers, при удалении записи из таблицы part_numbers соответствующая

			запись из таблицы plans также будет удалена
amount	int unsigned		Не может быть NULL
is_prioritize	boolean		Не может быть NULL

Таблица 7 – «seats_remains» – хранит остатки от плана для каждого рабочего места

Наименование поля	Тип данных	Признак ключа	Примечание
seats_remains_id	int unsigned	PK	Уникальное значение
plan_id	int unsigned	FK	Не может быть NULL, ссылается на поле plan_id таблицы plans, при удалении записи из таблицы plans соответствующая запись из таблицы seats_remains также будет удалена
remain	int unsigned		Не может быть NULL
seat_number	int unsigned	FK	Не может быть NULL, ссылается на поле seat_number таблицы seats, при удалении записи из таблицы seats соответствующая запись из таблицы

			seats_remains также будет удалена
--	--	--	--------------------------------------

Таблица 8 – «component_names» – хранит названия компонентов

Наименование поля	Тип данных	Признак ключа	Примечание
component_name_id	int unsigned	РК	Уникальное значение
component_name	varchar(50)		Не может быть NULL

Таблица 9 – «components» – хранит информацию о компонентах

Наименование поля	Тип данных	Признак ключа	Примечание
component_id	int unsigned	РК	Уникальное значение
component_name_id	int unsigned	FK	Не может быть NULL, ссылается на поле component_name_id таблицы component_names, при удалении записи из таблицы components_names соответствующая запись из таблицы components также будет удалена
component_number	varchar(50)		Не может быть NULL
amount	int unsigned		

Таблица 10 – «components_seats» – хранит информацию об используемых компонентах на местах

Наименование поля	Тип данных	Признак ключа	Примечание
components_seats_id	int unsigned	PK	Уникальное значение
component_id	int unsigned	FK	Не может быть NULL, ссылается на поле component_id таблицы components, при удалении записи из таблицы components соответствующая запись из таблицы components_seats также будет удалена
seat_number	int unsigned	FK	Не может быть NULL, ссылается на поле seat_number таблицы seats, при удалении записи из таблицы components соответствующая запись из таблицы components_seats также будет удалена
max_amount	int unsigned		Не может быть NULL

Таблица 11 – «components_variations» – хранит принадлежность компонентов к исполнениям

Наименование поля	Тип данных	Признак ключа	Примечание
components_variations_id	int unsigned	PK	Уникальное значение
part_number_id	int unsigned	FK	Не может быть NULL, ссылается на поле part_number_id

			таблицы part_numbers, при удалении записи из таблицы part_numbers соответствующая запись из таблицы components_variations также будет удалена
component_id	int unsigned	FK	Не может быть NULL, ссылается на поле component_id таблицы components, при удалении записи из таблицы components соответствующая запись из таблицы components_variations также будет удалена

Таблица 12 – «defectivity_types» – хранит типы брака

Наименование поля	Тип данных	Признак ключа	Примечание
defectivity_type_id	int unsigned	PK	Уникальное значение
defectivity_type	varchar(12)		Не может быть NULL

Таблица 13 – «defectivity» – хранит информацию о браке

Наименование поля	Тип данных	Признак ключа	Примечание
defectivity_id	int unsigned	PK	Уникальное значение
defectivity_date	date		Не может быть NULL

defectivity_time	time		Не может быть NULL
defectivity_type_id	int unsigned	FK	Не может быть NULL, ссылается на поле defectivity_type_id таблицы defectivity_types, при удалении записи из таблицы defectivity_types соответствующая запись из таблицы defectivity также будет удалена
defectivity_component_id	int unsigned	FK	Не может быть NULL, ссылается на поле component_id таблицы components, при удалении записи из таблицы components соответствующая запись из таблицы defectivity также будет удалена
amount	int unsigned		Не может быть NULL
record_ok	boolean		

Таблица 14 – «tf_1» – хранит информацию об операции на первом рабочем месте

Наименование поля	Тип данных	Признак ключа	Примечание
-------------------	------------	---------------	------------

tf_1_id	int unsigned	PK	Уникальное значение
operator_card_number	int unsigned	FK	Не может быть NULL, ссылается на поле card_number таблицы persons, при удалении записи из таблицы persons соответствующая запись из таблицы tf_1 также будет удалена
scan_date	date		Не может быть NULL
scan_time	time		Не может быть NULL
scanned_info	text		
scan_info_writed_date	date		
scan_info_writed_time	time		
is_success	boolean		
record_ok	boolean		

Таблица 15 – «logs_types» – хранит типы логов

Наименование поля	Тип данных	Признак ключа	Примечание
log_type_id	int unsigned	PK	Уникальное значение
log_type	varchar(30)		Не может быть NULL

Таблица 16 – «logs» – хранит все логи

Наименование поля	Тип данных	Признак ключа	Примечание
log_id	int unsigned	PK	Уникальное значение

log_date	date		Не может быть NULL
log_time	time		Не может быть NULL
card_number	int unsigned		Не может быть NULL
seat_number	int unsigned	FK	Не может быть NULL, ссылается на поле seat_number таблицы seats, при удалении записи из таблицы seats соответствующая запись из таблицы logs также будет удалена
log_type	int unsigned		Не может быть NULL
log_description	text		Не может быть NULL
record_ok	boolean		

2.3. Структура десктоп-приложения

Функционально ПО для рабочих мест сборки состоит из:

- идентификации оператора
 - определение уровня прав
 - прозрачность производства
 - подпись под этапом над изделием
- совершение операции
 - учет компонентов
 - совершение операций оператором
 - выполнение плана

- процедура брака
 - брак производства
 - брак поставщика
- управление пользователем на этом рабочем месте
 - добавление пользователя
 - изменение уровня прав
 - снятие оператора с рабочего места

Программная составляющая ПО рабочего места сборки:

- MainForm.cs – работа с главным потоком UI формы, обработчики событий взаимодействия пользователя и формы, главный файл
- ScannerForm.cs – форма для настройки сканера, представляет из себя фотографии штрих-кодов для настройки
- Database.cs – класс для работы с базой данных
- CardReader.cs – класс для работы со считывателем карт по COM-порту
- IniFile.cs – класс для работы с конфигурационным файлом
- Scanner.cs – класс для работы со сканером баркодов по виртуальному COM-порту
- WinShutdown.cs – класс для выключения системы
- PersonHelpForm.dll – динамическая библиотека классов, предоставляющая форму для работы с персоналом для текущего рабочего места
- LogsDatabase.dll - динамическая библиотека классов, необходимая для логирования действий оператора
- ArchiveTool.dll - динамическая библиотека классов, необходимая для распаковки запароленного архива с конфигурационным файлом

Функционально ПО-конструктор для рабочих мест сборки состоит из:

- создать рабочие место
 - создать список компонентов для рабочего места
- изменить рабочее место
 - добавить компонент к рабочему месту

- снять компонент с рабочего места
- заменить компонент
- изменить используемое количество на рабочем месте
- работа с компонентами
 - добавление компонента
 - удаление компонента
 - добавить имеющееся количество компонентов
- работа с планом
 - создать обычный план
 - создать приоритетный план

Программная составляющая ПО-конструктора:

- MainForm.cs – работа с главным потоком UI формы, обработчики событий взаимодействия пользователя и формы, главный файл
- Database.cs – класс для работы с базой данных
- CardReader.cs – класс для работы со считывателем карт по COM-порту
- IniFile.cs – класс для работы с конфигурационным файлом
- LogsDatabase.dll - динамическая библиотека классов, необходимая для логирования действий оператора
- ArchiveTool.dll - динамическая библиотека классов, необходимая для распаковки запароленного архива с конфигурационным файлом

Выводы

В результате разработки структуры автоматизированной системы технологического процесса сборки тахографа была спроектирована модульная архитектура, включающая ключевые компоненты: авторизацию, управление пользователями, конструктор техпроцесса, управление сборкой, взаимодействие с оборудованием, контроль качества и отчетность.

Предложенная структура обеспечивает:

- гибкость и расширяемость системы за счет модульного подхода;

- возможность точного контроля действий операторов и качества продукции;
- централизованное хранение и обработку информации в базе данных;
- удобную интеграцию с оборудованием и периферийными устройствами через COM-порты;
- надежную регистрацию всех операций сборки и отклонений от техпроцесса.

Данная структура является прочной основой для реализации функциональности системы и обеспечивает выполнение требований к надежности, безопасности и контролю качества на всех этапах сборки тахографа.

3. ТЕСТИРОВАНИЕ И ИНТЕГРАЦИЯ КОМПОНЕНТОВ ПРОГРАММНОГО ПРОДУКТА

3.1. Тестирование системы

Целью тестирования является проверка работоспособности, стабильности и соответствия разработанной системы автоматизации технологического процесса сборки тахографа установленным требованиям. Основными задачами тестирования являются:

- проверка корректности выполнения функций согласно техническому заданию;
- оценка устойчивости системы к ошибочным действиям пользователя;
- проверка работы системы с подключённым оборудованием через COM-порт;
- оценка пользовательского интерфейса на предмет удобства и логичности;
- проверка работы модуля конструктора технологического процесса;
- тестирование обработки и хранения данных в базе данных MySQL.

Тестирование проводилось вручную на тестовом стенде, имитирующем реальное рабочее место оператора. Были смоделированы различные сценарии использования системы, включая:

- авторизацию пользователя с различными правами;
- запуск и выполнение этапов технологического процесса;
- обмен данными с оборудованием через COM-порт;
- управление пользователями и настройками системы;
- добавление, редактирование и удаление шаблонов технологического процесса.

Кроме того, выполнялось тестирование работы с базой данных: проверка корректного сохранения информации, работы с транзакциями и восстановления данных при сбоях соединения.

Требования к десктоп-приложению

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

- Процессор: не ниже Intel Core i3;
- Оперативная память: не менее 4 ГБ;
- Свободное место на диске: не менее 500 МБ;
- Наличие COM-порта или адаптера USB-to-COM;
- Поддержка подключения оборудования;
- Операционная система: Windows 10 или выше;
- .NET Framework 4.7.2 или выше;
- Установленная MySQL Server 5.7 или выше;
- Visual C++ Redistributable;
- COM-драйвер для используемого оборудования.

Требования к программному обеспечению

Для работы приложения необходимо возможность полноценного подключения к виртуальному серверу предприятия.

3.2. Руководство администратора

Запуск и поддержка ПО-конструктора

Для запуска рабочего места необходимо наличие установленных средств для работы с языком программирования C#, .NET Framework и WinForms. Из аппаратных средств необходим лишь считыватель рабочих пропусков для идентификации рабочего (для данного ПО необходим самый высокий уровень прав).

Необходимо стабильное подключение к серверу предприятия, так как возможные изменения необходимо зафиксировать в базе данных, которая является централизованной для всех рабочих мест сборки.

Для начала работы необходимо убедиться, что оператор имеет соответствующий уровень прав, в ином случае оператор не допущен к работе.

Попытка о попытке несанкционированной работы будет зафиксирована в базе данных.

Место бригадира

ID: 16662831 Пользователь: Губин Егор Вячеславович

Создать план | Компоненты на местах | Компоненты

Исполнение:

Количество:

Приоритетный: ☐

Создать план

Рис. 3.1 Форма создания плана

Место бригадира

ID: 16662831 Пользователь: Губин Егор Вячеславович

Создать план | Компоненты на местах | Компоненты

Рабочее место	Мин. время работы (в секундах)
№ 1: Приклеивание (стекла к панели)	10
№ 2: Свинчивание (блок индикации и управления)	10
№ 3: Свинчивание (термопринтер)	10
№ 4: Программирование (запись загрузчика)	10
№ 5: Свинчивание (установка считывателей и платы)	10
№ 6: Сборка (установка лицевой панели)	10
№ 7: Запрессовка пружин	10
№ 8: Программирование внешнего модуля	10
№ 9: КФ внешнего модуля	10
№ 10: Программирование (запись ПО)	10
№ 11: Свинчивание	10
№ 12: Контроль функционирования (КФ)	10

Создать РМ | Удалить РМ | Открыть РМ | Изменить время

Обновить таблицу | Изменить № РМ | Свап РМ

Рис. 3.2 Форма со списком всех рабочих мест

Место бригадира

ID: 16662831 Пользователь: Губин Егор Вячеславович

Создание рабочего места

Название:

Мин. время операции (в секундах):

Пароль админа:

Номер места

☒ Ввести номер вручную

☐ Выбрать после какого РМ добавить

Рис. 3.3 Форма создания рабочего места

Место бригадира

ID: 16662831 Пользователь: Губин Егор Вячеславович

Компоненты на РМ № 2

Компонент	Используемое количество	Исполнения
Блок индикации и управления 21.3840510-...	1	-
Кнопка 21.3840530-10	1	21.3840000-10 21.3840000-11
Кнопка 21.3840530-01-10	1	21.3840000-10 21.3840000-11
Кнопка 21.3840530-02-10	1	21.3840000-10 21.3840000-11
Кнопка 21.3840530-03-10	1	21.3840000-10 21.3840000-11
Кнопка 21.3840530-04-10	1	21.3840000-10 21.3840000-11

Рис. 3.4 Форма конфигурирования рабочего места

Место бригадира

ID: 16662831 Пользователь: Губин Егор Вячеславович

Замена компонента Кнопка кассеты принтера 21.3840521-10Д. Выберите заменяющий

Компонент	Вариации	Количество
▸ DIN562 Гайка квадратная низкая М3	-	0
DIN7981 Саморез с полукруглой головкой ...	-	0
DIN7981 Саморез с полукруглой головкой ...	-	0
DIN7985 Винт с полукруглой головкой PH...	-	0
Блок индикации и управления 21.3840510-...	-	0
Заглушка USB 21.3840502-10	-	0
Кабельная сборка GKZS-FAKRACZG-RG17...	-	0
Кнопка 21.3840530-10	21.3840000-10 21.3840000-11	0
Кнопка 21.3840530-01-10	21.3840000-10 21.3840000-11	0
Кнопка 21.3840530-02-10	21.3840000-10 21.3840000-11	0
Кнопка 21.3840530-03-10	21.3840000-10 21.3840000-11	0

Заменить Назад

Рис. 3.5 Форма замены компонента на рабочем месте

Место бригадира

ID: 16662831 Пользователь: Губин Егор Вячеславович

Создать план Компоненты на местах Компоненты

Компонент	Исполнение	Количество
▸ DIN562 Гайка квадратная низкая М3	-	0
DIN7981 Саморез с полукруглой го...	-	0
DIN7981 Саморез с полукруглой го...	-	0
DIN7981 Саморез с полукруглой го...	-	0
DIN7985 Винт с полукруглой головк...	-	0
Блок индикации и управления 21.38...	-	0
Держатель рулона 21.3840522-10	21.3840000-10 21.3840000-11	0
Держатель рулона 21.3840522-10Д	21.3840000-12 21.3840000-13	0
Заглушка USB 21.3840502-10	-	0
Кабельная сборка GKZS-FAKRACZ...	-	0
Кнопка 21.3840530-10	21.3840000-10 21.3840000-11	0

Создать Изменить Удалить Добавить кол-во Обновить таблицу

Рис. 3.6 Форма всех компонентов для технологического процесса

Место бригадира

ID: 16662831 Пользователь: Губин Егор Вячеславович

Создание компонента

Название:

Номенклатура:

Количество:

Переменная: ☒

Исполнение:

Вариации
21.3840000-10
21.3840000-11
21.3840000-12
21.3840000-13

Рис. 3.7 Форма создания нового компонента

Место бригадира

ID: 16662831 Пользователь: Губин Егор Вячеславович

Изменение компонента

Название:

Номенклатура:

Количество:

Переменная: ☒

Исполнение:

Вариации
21.3840000-10
21.3840000-11
21.3840000-12
21.3840000-13

Рис. 3.8 Форма изменения компонента

Место бригадира

ID: 16662831 Пользователь: Губин Егор Вячеславович

Создать план Компоненты на местах Компоненты

Рабочее место	Мин. время работы (в секундах)
№ 1: Приклеивание (стекла к панели)	10
№ 2: Свинчивание (блок индикации и управления)	10
№ 3: Свинчивание (термопринтер)	
№ 4: Программирование (запись загрузочной записи)	
№ 5: Свинчивание (установка считывателя)	
№ 6: Сборка (установка лицевой панели)	
№ 7: Запрессовка пружин	
№ 8: Программирование внешнего модуля	10
№ 9: КФ внешнего модуля	10
№ 10: Программирование (запись ПО)	10
№ 11: Свинчивание	10
№ 12: Контроль функционирования (КФ)	10

Введите время

Введите время на операцию в секундах:

Подтвердить

Создать РМ Удалить РМ Открыть РМ Изменить время

Обновить таблицу Изменить № РМ Свал РМ

Рис. 3.9 Форма изменения времени на операцию на рабочем месте

3.3. Руководство пользователя

Руководство к десктоп-приложению

Для запуска рабочего места необходимо наличие установленных средств для работы с языком программирования C#, .NET Framework и WinForms. Из аппаратных средств необходимы считыватель рабочих пропусков для идентификации рабочего, Bluetooth в виде «USB-свистка», сканер баркодов для считывания информации с этикеток.

Необходимо стабильное подключение к серверу предприятия, на котором находится централизованная для технологического процесса база данных.

Для начала работы необходимо убедиться, что оператор имеет соответствующий уровень прав, в ином случае оператор не допущен к работе. Попытка о попытке несанкционированной работы будет зафиксирована в базе данных.

Перед началом работы необходимо пройти обучающий курс, интегрированный в ПО.

Рабочее место Т-ф2

Сканер Персонал

Элемент	
▶	Переменные
	Кнопка 21.3840530-10
	Кнопка 21.3840530-01-10
	Кнопка 21.3840530-02-10
	Кнопка 21.3840530-03-10
	Кнопка 21.3840530-04-10
	Кнопка 21.3840530-05-10
	Кнопка 21.3840530-10Д
	Кнопка 21.3840530-01-10Д
	Кнопка 21.3840530-02-10Д

Рис. 3.10 Форма до начала работы, необходима для контроля компонентов на рабочем месте

Рабочее место Т-ф2

Сканер Персонал

ID:	16662831	ФИО:	Губин Егор												
Исполнение:	21.3840000-	Количество:	1200 шт.												
Переменные Постоянные Обучение															
<table border="1"> <thead> <tr> <th colspan="2">Элемент</th> </tr> </thead> <tbody> <tr> <td>▶</td> <td>Кнопка 21.3840530-10Д</td> </tr> <tr> <td></td> <td>Кнопка 21.3840530-01-10Д</td> </tr> <tr> <td></td> <td>Кнопка 21.3840530-02-10Д</td> </tr> <tr> <td></td> <td>Кнопка 21.3840530-03-10Д</td> </tr> <tr> <td></td> <td>Кнопка 21.3840530-04-10Д</td> </tr> </tbody> </table>				Элемент		▶	Кнопка 21.3840530-10Д		Кнопка 21.3840530-01-10Д		Кнопка 21.3840530-02-10Д		Кнопка 21.3840530-03-10Д		Кнопка 21.3840530-04-10Д
Элемент															
▶	Кнопка 21.3840530-10Д														
	Кнопка 21.3840530-01-10Д														
	Кнопка 21.3840530-02-10Д														
	Кнопка 21.3840530-03-10Д														
	Кнопка 21.3840530-04-10Д														
Брак поставщика		Брак													
		Успешная													

Рис. 3.11 Форма переменных компонентов, зависящих от исполнения

Рабочее место Т-ф2

Сканер Персонал

ID:	16662831	ФИО:	Губин Егор					
Исполнение:	21.3840000-	Количество:	1200 шт.					
Переменные Постоянные Обучение								
<table border="1"> <thead> <tr> <th>Элемент</th> </tr> </thead> <tbody> <tr> <td>Блок индикации и управления 21.3840510-10</td> </tr> <tr> <td>Пружина 21.3840504-10</td> </tr> <tr> <td>DIN7981 Саморез с полукруглой головкой PH d2.9x9.5 Ц...</td> </tr> <tr> <td>Шлейф JK_L-KLS17-FFC-0.5-22P-L030A4/8_KLS</td> </tr> </tbody> </table>				Элемент	Блок индикации и управления 21.3840510-10	Пружина 21.3840504-10	DIN7981 Саморез с полукруглой головкой PH d2.9x9.5 Ц...	Шлейф JK_L-KLS17-FFC-0.5-22P-L030A4/8_KLS
Элемент								
Блок индикации и управления 21.3840510-10								
Пружина 21.3840504-10								
DIN7981 Саморез с полукруглой головкой PH d2.9x9.5 Ц...								
Шлейф JK_L-KLS17-FFC-0.5-22P-L030A4/8_KLS								
Брак поставщика		Брак						
		Успешная						

Рис. 3.12 Форма постоянных компонентов

Рабочее место Т-ф2

Сканер Персонал

ID:	16662831	ФИО:	Губин Егор																								
Исполнение:	21.3840000-	Количество:	1200 шт.																								
<table border="1"> <thead> <tr> <th>Элемент</th> <th>-</th> <th>Кол-во</th> <th>+</th> </tr> </thead> <tbody> <tr> <td colspan="4">Переменные</td> </tr> <tr> <td>Кнопка 21.3840530-10Д</td> <td>-</td> <td>0</td> <td>+</td> </tr> <tr> <td>Кнопка 21.3840530-01-10Д</td> <td>-</td> <td>1</td> <td>+</td> </tr> <tr> <td>Кнопка 21.3840530-02-10Д</td> <td>-</td> <td>1</td> <td>+</td> </tr> <tr> <td>Кнопка 21.3840530-03-10Д</td> <td>-</td> <td>0</td> <td>+</td> </tr> </tbody> </table>				Элемент	-	Кол-во	+	Переменные				Кнопка 21.3840530-10Д	-	0	+	Кнопка 21.3840530-01-10Д	-	1	+	Кнопка 21.3840530-02-10Д	-	1	+	Кнопка 21.3840530-03-10Д	-	0	+
Элемент	-	Кол-во	+																								
Переменные																											
Кнопка 21.3840530-10Д	-	0	+																								
Кнопка 21.3840530-01-10Д	-	1	+																								
Кнопка 21.3840530-02-10Д	-	1	+																								
Кнопка 21.3840530-03-10Д	-	0	+																								
Забраковать		Назад																									
Брак поставщика		Брак																									
		Успешная																									

Рис. 3.9 Форма для процедуры брака

Выводы

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

ЗАКЛЮЧЕНИЕ

В ходе выполнения курсовой работы была разработана система автоматизации технологического процесса сборки тахографа, предназначенная для повышения эффективности управления сборочным процессом, контроля качества и сокращения количества брака.

В процессе проектирования были выполнены следующие задачи:

- проведён анализ предметной области и обоснована необходимость автоматизации;
- сформулированы функциональные и нефункциональные требования к системе;
- разработана структура базы данных и реализована связь с СУБД MySQL;
- создано десктопное приложение на языке программирования C# с использованием WinForms;
- реализована работа с оборудованием, подключаемым через COM-порт;
- спроектирован и реализован модуль конструктора технологического процесса;
- введена система учётных записей и прав доступа пользователей.

Разработанное программное обеспечение соответствует требованиям, изложенным в техническом задании, и обеспечивает все заявленные функции: создание, редактирование и хранение технологических процессов, работа с оборудованием, контроль выполнения операций, учёт результатов и авторизацию пользователей.

Система была протестирована в различных режимах работы, в том числе при взаимодействии с внешними устройствами. По результатам тестирования можно сделать вывод о корректной и стабильной работе приложения.

Таким образом, цель курсовой работы достигнута: созданное программное обеспечение может быть использовано в условиях реального производства для автоматизации сборки тахографов, улучшения контроля качества и повышения общей производственной дисциплины.

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. Богачёв, А. В. Проектирование баз данных [Текст]: учебное пособие / А. В. Богачёв. — М.: Форум, 2019. — 224 с.

2. Вайнштейн, М. З. Основы научных исследований [Электронный ресурс]: учебное пособие / М. З. Вайнштейн, В. М. Вайнштейн, О. В. Кононова. — Йошкар-Ола: Марийский государственный технический университет, Поволжский государственный технологический университет, ЭБС АСВ, 2011. — 216 с. — Режим доступа: <http://www.iprbookshop.ru/22586.html>
3. Головастики С. Н. Программирование в среде .NET [Текст]: учебное пособие / С. Н. Головастики. — СПб.: БХВ-Петербург, 2021. — 352 с.
4. Документация C# [Электронный ресурс]. — Microsoft Learn. — Режим доступа: <https://learn.microsoft.com/ru-ru/dotnet/csharp/>
5. Документация MySQL [Электронный ресурс]. — Oracle. — Режим доступа: <https://dev.mysql.com/doc/>
6. Документация WinForms [Электронный ресурс]. — Microsoft Learn. — Режим доступа: <https://learn.microsoft.com/ru-ru/dotnet/desktop/winforms/>
7. Дьяконов В. Ю. Программирование баз данных в Visual Studio [Текст] / В. Ю. Дьяконов. — СПб.: Питер, 2022. — 304 с.
8. Кайзер Д. Программирование на C# для начинающих. — Пер. с англ. — М.: ДМК Пресс, 2020. — 416 с.
9. Майер, Б. Объектно-ориентированное программирование и программная инженерия [Электронный ресурс] / Б. Майер. — 3-е изд. — М.: ИНТУИТ, Ай Пи Эр Медиа, 2019. — 285 с. — Режим доступа: <http://www.iprbookshop.ru/79706.html>
10. Маюрникова, Л. А. Основы научных исследований в научно-технической сфере [Электронный ресурс]: учебно-методическое пособие / Л. А. Маюрникова, С. В. Новосёлов. — Кемерово: КемТИПП, 2009. — 123 с. — Режим доступа: <http://www.iprbookshop.ru/14381.html>
11. Моделирование систем [Текст]: учебник для вузов / С. И. Дворецкий, Ю. Л. Муромцев, В. А. Погонин, А. Г. Схиртладзе. — М.: Академия, 2009. — 320 с.
12. Щербаков, А. Интернет-аналитика [Электронный ресурс]: поиск и оценка информации в web-ресурсах: практическое пособие / А. Щербаков. — М.:

Книжный мир, 2012. — 78 с. — Режим доступа:
<http://biblioclub.ru/index.php?page=book&id=89693>