

Факультет информатики и робототехники  
Кафедра технической кибернетики

**РАСЧЕТНО-ПОЯСНИТЕЛЬНАЯ ЗАПИСКА  
К ВЫПУСКНОЙ КВАЛИФИКАЦИОННОЙ РАБОТЕ  
по направлению 09.03.01 Информатика и вычислительная  
техника, профиль «Программное обеспечение средств ВТ и АС»**  
(шифр, наименование)

НА ТЕМУ: Модуль генерации штрихкодов для ERP-системы

К защите допущен

Зав. кафедрой  
Бежаева О.Я. (\_\_\_\_\_) (\_\_\_\_\_) (подпись)  
(фамилия, инициалы)

« 23 » 06 2023 г.

Обучающийся

Гусманов А. И. (\_\_\_\_\_) (\_\_\_\_\_) (подпись)  
(фамилия, инициалы)

Руководитель выпускной квалификационной работы

Бежаева О.Я. (\_\_\_\_\_) (\_\_\_\_\_) (подпись)  
(фамилия, инициалы)

# ФГБОУ ВО «Уфимский университет науки и технологий»

Факультет информатики и робототехники  
Кафедра технической кибернетики

«УТВЕРЖДАЮ»  
Зав. кафедрой \_\_\_\_\_/Бежаева О. Я.  
(подпись, Фамилия И.О.)  
« 03 » \_\_\_\_\_ апреля 2023 г.

## ЗАДАНИЕ

### по подготовке выпускной квалификационной работы

обучающемуся Гусманову Алмазу Илдусовичу группы ИБТ-421  
(фамилия, имя, отчество)

1 Тема выпускной квалификационной работы: Модуль генерации штрихкодов для ERP-системы

(утверждена приказом от 01.03.2023 № 0446)

2. Срок сдачи обучающимся законченной выпускной квалификационной работы 22.06.2023

3. Исходные данные к выпускной квалификационной работе пользовательские требования, существующая информационная система предприятия

4. Перечень вопросов, подлежащих разработке в выпускной квалификационной работе (краткое содержание, при необходимости с указанием разделов) Анализ предметной области использования технологии штрихкодирования; Проектирование моделей бизнес-процесса; Разработка модуля генерации штрихкодов для ERP-системы

[illegible]

5. Цель и объем патентных исследований Не предусмотрено

6. Объем и степень использования программного обеспечения Microsoft Word, Microsoft PowerPoint, Microsoft Visio, ERwin Data Modeler, Rational Rose, MySQL, Visual Studio Code

7. Объем расчетно-пояснительной записки на 69 листах А4 формата.

Дата выдачи задания « 31 » 03 20 23 г.

Руководитель \_\_\_\_\_

« 31 » 03 20 23 Г.

**ФГБОУ ВО «Уфимский университет науки и технологий»**

Факультет информатики и робототехники

Кафедра технической кибернетики

«УТВЕРЖДАЮ»

Зав. кафедрой \_\_\_\_\_/Бежаева О.Я. \_  
(подпись, Фамилия И.О.)

« 23 » \_\_\_\_\_ 04 \_\_\_\_\_ 2023 г.

**КАЛЕНДАРНЫЙ ПЛАН**

**работы над выпускной квалификационной работой**

обучающегося: Гусманова Алмаза Илдусовича \_\_\_\_\_ группы ИБТ-421  
(фамилия, имя, отчество)

Тема выпускной квалификационной работы: Модуль генерации штрихкодов для ERP-системы

№ п/п	Наименование разделов выпускной квалификационной работы	Срок	Объем (в % от всей выпускной квалификационной работы)	Фактическое (объем работы в %)
<b>Расчет и описание</b>				
1	Анализ предметной области использования технологии штрихкодирования	13.04.2023	37	35
2	Проектирование моделей бизнес-процесса	28.05.2023	30	32
3	Разработка модуля генерации штрихкодов для ERP-системы	13.06.2023	33	33

Обучающийся \_\_\_\_\_  
(подпись)

« 31 » \_\_\_\_\_ 03 \_\_\_\_\_ 2023 г.

Руководитель выпускной квалификационной работы \_\_\_\_\_  
(подпись)

« 31 » \_\_\_\_\_ 03 \_\_\_\_\_ 2023 г.

Ход выполнения выпускной квалификационной работы:

Дата просмотра выпускной квалификационной работы на кафедре	13.04.23	28.05.23	13.06.23	22.06.23
Объем выполнения выпускной квалификационной работы в %	30	60	80	100

Дата защиты выпускной квалификационной работы на заседании ГЭК « 26 » 06 2023 г.

## ПАМЯТКА ДИПЛОМНИКУ

### Дипломник должен:

1. Придерживаться следующей рекомендуемой последовательности подшивки документов:
  - титульный лист;
  - задание по подготовке выпускной квалификационной работы;
  - календарный план;
  - аннотация;
  - содержание;
  - введение;
  - разделы ВКР (в соответствии с требованиями к написанию выпускной квалификационной работы по направлению (специальности))
  - заключение;
  - список литературы;
  - приложения;
  - отзыв руководителя;
  - справка о результатах проверки ВКР на плагиат

Первым листом пояснительной записки считается титульный лист. Нумерацию листов выпускной квалификационной работы начинать с листа, на котором располагается основная надпись.

Для защиты выпускной квалификационной работы в установленный срок обучающийся должен:

2. Представить выпускную квалификационную работу в срок не позднее 13.06.2023 на предварительный просмотр на кафедру (при наличии такой процедуры). Результат предварительной защиты выпускной квалификационной работы отражается в специальном Акте комиссии, назначаемой кафедрой.
3. Представить готовую выпускную квалификационную работу в срок не позднее 22.06.2023 на подпись руководителю.
4. Представить выпускную квалификационную работу на рецензию.
5. Представить выпускную квалификационную работу на подпись заведующему кафедрой.
6. Передать на кафедру анкетные данные консультанта выпускной квалификационной работы и рецензента (на специальном бланке), согласие на обработку персональных данных.

## АННОТАЦИЯ

Пояснительная записка 74 с., 17 рис., 10 табл., 36 источников, 1 прил.

### ГЕНЕРАЦИЯ, ШТРИХКОД, МОДУЛЬ, *ERP*-СИСТЕМА

Данная выпускная квалификационная работа посвящена разработке модуля генерации штрихкодов для *ERP*-системы.

Проведен анализ предметной области и выявлены проблемы данных областей.

Спроектированы бизнес-модели процессов при помощи нотации *BPMN*, а также представлены в виде мнемосхемы.

Разработаны требования к создаваемой системе.

Спроектирован прототип интерфейса модуля. По прототипу разработан модуль генерации штрихкодов для *ERP*-системы.

# СОДЕРЖАНИЕ

ВВЕДЕНИЕ.....	6
1 Анализ предметной области использования технологии штрихкодирования .....	8
1.1 Актуальность .....	8
1.1.1 Система штрихового кодирования .....	8
1.1.2 Штрихкодирование в складском учете .....	9
1.1.3 Штрихкодирование в адресном хранении .....	11
1.1.4 Штрихкодирование в области делопроизводства.....	15
1.1.5 Штрихкодирование в обеспечении безопасности .....	17
1.1.6 <i>ERP</i> -система.....	18
1.2 Виды штрихкодов.....	26
1.3 Цель и задачи выпускной квалификационной работы.....	28
2 Проектирование моделей бизнес-процесса .....	29
2.1 Требования к программному продукту.....	29
2.2 Выбор методологии моделирования бизнес-процесса.....	30
2.2.1 Описание методологии <i>BPMN</i> .....	30
2.2.2 Разработка модели функционирования бизнес-процесса .....	32
2.3 Описание логической и физической модели данных .....	36
2.4 Алгоритм работы программы .....	42
2.5 Архитектура системы .....	43

					<b>3271.102092.000 ПЗ</b>		
Изм.	Лист	№ докум.	Подпись	Дата			
Разраб.		Гусманов А.И.			Модуль генерации штрихкодов для ERP-системы Пояснительная записка		
Руковод.		Бежаева О.Я.					
Рецензент.							
Н. Контр.		Мухаметьянова Р.И.					
Утв.		Бежаева О.Я.					
					Лит.	Лист	Листов
					У	5	74
					ФГБОУ ВО «УУНУТ», ИБТ-421		

3 Разработка модуля генерации штрихкодов для ERP-системы .....	44
3.1 Используемые технологии .....	44
3.1.1 <i>REST API</i> .....	44
3.1.2 Модель <i>OSI</i> .....	48
3.1.3 <i>JavaScript, HTML, CSS</i> .....	50
3.2 Руководство пользователя.....	57
3.3 Руководство программиста .....	60
3.3.1 Режим работы программы .....	60
3.3.2 Средства проверки правильности выполнения программы .....	60
3.3.3 Входные и выходные данные.....	61
3.3.4 Описание основных функций .....	62
СПИСОК ЛИТЕРАТУРЫ.....	67
ПРИЛОЖЕНИЕ А .....	70

Инв. № подл.	Подп. и дата	Инв. № дубл.	Взам. инв. №	Подп. и дата						
Изм.	Лист	№ докум.	Подпись	Дата	3271.102092.000 ПЗ					Лист
										6



## ВВЕДЕНИЕ

В современном мире система штрихового кодирования используется повсеместно от самозанятых предпринимателей до крупных корпораций. Данная технология зарекомендовала себя как эффективный способ кодирования данных в графическом виде. В связи с распространенностью данной технологии, повышается необходимость в углубленном изучении темы дипломного проекта.

Крупные торговые организации используют данную технологию для преобразования имеющихся данных о товарах. Благодаря этому продавцы получили широкий диапазон возможностей в совершении товарных операций, отслеживании товарооборота в разных сферах, оптимизации складского учета.

Организации интегрируют *ERP*-системы для автоматизации и оптимизации бизнес-процессов. В данных информационных системах необходим инструмент для генерации и печати штриховых кодов.

Данная работа направлена на создание модуля по генерации штрихкодов для *ERP*-систем. Пользователю предоставляется возможность получить штрихкод по введенным данным в необходимом формате и передать его на печать.

Цель выпускной квалификационной работы:

- повышение эффективности складского учета путем использования модуля генерации штрихкодов для *ERP*-системы.

Для достижения данной цели необходимо выполнение следующих задач:

- проведение анализа предметной области;
- проведение анализа предлагаемого процесса;
- проектирование моделей предлагаемого бизнес-процесса;
- разработка моделей базы данных;
- проектирование модели архитектуры системы;
- создание модуля для генерации штрихкодов и их печати.

Для решения данных задач использовались теоретические и практические методы исследования.

Инв. № подл.	Подп. и дата	Инв. № дубл.	Взам. инв. №	Подп. и дата
Изм.	Лист	№ докум.	Подпись	Дата

3271.102092.000 ПЗ

Лист 7



маркировки «Честный знак», в случае продажи и распространения товаров из списка категории для обязательной маркировки, производитель столкнется с использованием штрихкодов формата «*Data Matrix*». Данный тип штрихкодов обеспечивает защиту передаваемых данных, так как создание фальсификации и подделки вшитой в код информации невозможно, вследствие применения технологий шифрования данных.

### 1.1.2 Штрихкодирование в складском учете

Данная технология нашла свое применение в реализации учета товаров на складах. В наше время все больше крупных торговых сетей или маленьких магазинов прибегают к реализации системы штрихкодирования товаров внутри собственной компании. Данная процедура необходима для оптимизации процессов приемки, выдачи, оприходования и многих других с помощью штрихкодовых обозначений. Данный вид передачи информации предоставляет корректную передачу данных о товаре без промедления, что в паре с существующей базой данных товаров компании предоставляет возможности для автоматизации учета и контроля продукции внутри компании. В случае отсутствия автоматизации учета товаров с использованием технологии штрихкодирования на производстве, складе или даже розничном магазине имеется большая вероятность возникновения ряда сложностей, таких как:

- образование проблем, спровоцированных совершением ошибок человеческого фактора;
- стагнация операционной деятельности;
- отсутствие возможностей в своевременном контроле и модернизации бизнес-процессов;
- издержки затрат имеющихся ресурсов;
- отсутствие актуальной отчетности о ходе развития бизнеса [36].

Процесс штрихкодирования товара включает в себя генерацию штрихкода по введенным данным о товаре, а также нанесение на продукцию или ее упаковку

Инв. № подл.	Подп. и дата	Инв. № дубл.	Взам. инв. №	Подп. и дата	<div>3271.102092.000 ПЗ</div>	Лист
						9
						Изм.

машиночитаемого штрихового кода. Данный процесс необходим для реализации учета товаров на складах и автоматизации существующих процессов приемки и выдачи, транспортировки изделия внутри предприятия и многих других. Развитие данного решения предоставляет возможности использования данных о товарах только на предприятиях компании, не сталкиваясь с проблемами утечек данных в другие организации. На товар наносится штрихкод, представляющий собой индивидуальный идентификатор продукции. Каждый из данных кодов может содержать в себе как базовую информацию о товаре, например – название, производитель, место производства, так и подробную информацию, такую как дату производства, срок годности, партия, спецификация и многие другие. Наличие или отсутствие в штрихкоде определенной информации регулируется руководством компании исходя из существующих задач. Также выбирается формат штрихкода, исходя из требований руководства, так как у каждого формата штрихкода имеются отличительные характеристики, которые могут не подходить для выполнения задач, например, размер штрихкода, технические возможности оборудования, количество сохраняемой информации и другие. Так, например, одномерный штрихкод формата *CODE-128* позволяет использовать до 48 символов, в отличие от двумерного штрихкода *QR-Code*, допустимое количество символов которого равно 7089 [3].

Преимущества внедрения технологии штрихкодирования на складе:

1) Исключение лишних запасов. Система отслеживания предоставляет информацию о наличии и количестве товаров в данный момент, что помогает в продаже товаров.

2) Рациональное использование техники. За каждым сотрудником закрепляется определенный вид техники во время использования. При помощи специальных считывающих устройств, расположенных по всему складу, имеется возможность для контроля и отслеживания в перемещениях любого аппарата.

3) Сокращение используемой площади склада при том же количестве товара. При помощи системы адресного хранения имеется возможность рационального

Инв. № подл.	Подп. и дата	Инв. № дубл.	Взам. инв. №	Подп. и дата						
					3271.102092.000 ПЗ					Лист
										10
Изм.	Лист	№ докум.	Подпись	Дата						

использования площади склада. Так, например, товары не расположены в разных местах, создавая порядок и определенность в использовании необходимой площади.

4) Сокращение затрат на аренду площади склада. Оптимизация при помощи системы адресного хранения позволяет сократить нерационально используемое количество площади. Чем меньше используется площади, тем меньше затрат на нее.

5) Повышение эффективности работы персонала. Контроль за складом и созданный порядок позволяют персоналу не затрачивать рабочие часы на поиск нужного товара в ходе различных процессов, связанных с продукцией.

6) Своевременное пресечение ошибок и сбоев в работе. Каждая ошибка отслеживается и сохраняется, что помогает сотрудникам компании в их исправлении.

7) Автоматизация оформления отчетности. Существующая база данных всех штрихкодов и товаров предоставляет возможность создания документов. Каждый атрибут и название товара передаются в модули делопроизводства для формирования отчетности [24].

Часто вместо с системой учета на складе внедряется система адресного хранения.

### 1.1.3 Штрихкодирование в адресном хранении

Ежедневное расширение и увеличение всемирного грузопотока и товарооборота влияет на модернизацию технологий и созданию практик по управлению ими. В настоящее время беспорядок в расстановке и бесконтрольная обстановка на складах временного размещения товаров должна отсутствовать, поскольку существуют запросы от арендаторов и клиентов на предоставление качественного и своевременного обслуживания. Поэтому технология штрихкодирования получила популярность в сфере хранения товаров, а именно в реализации адресной системы хранения на складе.

Инв. № подл.	Подп. и дата
Инв. № дубл.	Взам. инв. №
Подп. и дата	Подп. и дата

Изм.	Лист	№ докум.	Подпись	Дата	3271.102092.000 ПЗ	Лист
						11

Всякий раз при поступлении товара на склад розничной сети или производителя, перед персоналом возникает проблема размещения поступившего товара. Для оптимизации процессов размещения товаров существует адресная система хранения.

Адресная система хранения товаров на складе — это способ организации складского пространства, при котором зона хранения разбивается на ячейки, и каждой из них присваивается код. Особенность данной системы в том, что благодаря использованию автоматизированных систем учета и управления, существенно сокращается время поиска любого товара на складе, что в свою очередь, уменьшает общее время сборки заказа. Данная система хорошо зарекомендовала себя в организациях, специализирующихся на доставке товаров в минимальные сроки, а также на складах с высоким грузооборотом, так как она обеспечивает необходимый уровень качества и скорости. Для организации ячеек существуют определенные правила и принципы, например, корректные размеры ячеек, границы ячеек должны быть конкретными, адрес ячейки должен быть разборчивым. Если склад автоматизирован, то в качестве дополнительной маркировки используются штрихкоды. Адрес ячейки строится по типу «От большего к меньшему». В случае работы на большом складе, то адрес начинается с символа корпуса, далее добавляется стеллаж и номер помещения, а в самом конце местоположение груза в этом помещении в соответствии с рисунком 1.



Рисунок 1 – Штрихкод для адресного хранения

Инв. № подл.	Подп. и дата
Инв. № дубл.	Взам. инв. №
Подп. и дата	Инв. № подл.
Инв. № подл.	Подп. и дата

Также существует ряд достоинств применения данного способа:

- сбор заказов происходит без задержек. Сотрудникам нет необходимости находится в поисках товара на огромных складах, затрачивая на это большое количество времени. Время работников имеет ценность как для самих работников, так и для работодателей, так как затраченное время прямо влияет на бюджетные издержки;

- происходит оптимизация большей части процессов, из-за этого персонал затрачивает меньше временных и физических ресурсов на затруднительные задачи и получает возможность вкладываться и работать в дополнительных сферах;

- ошибки человеческого фактора минимизируются, так как оптимизируется большинство процессов, которые нет необходимости выполнять персоналу самостоятельно;

- ускорение процесса инвентаризации, вследствие упрощения проведения процесса пересчета товара, оприходования;

- человек перестает быть единственным знатоком помещений. До введения системы хранения только опытный сотрудник знал, где лежит конкретный товар [28].

Существует несколько видов адресного хранения: статический и динамический.

Статическое адресное хранение предполагает закрепление конкретного адреса за определенным товаром или группы адресов за группой товаров. Это означает, что будет выделяться определенное место, на которое и будет размещаться данный товар или его аналоги и никакой больше. Например, на обувном складе для каждого вида обуви используется определенный стеллаж. Стеллаж делится на отсеки исходя из характеристик данного вида товара.

Преимущества данного подхода:

- прозрачность и последовательность процессов;
- подбор продукции без промедлений;
- обучаемость персонала в требуемый срок.

Инв. № подл.	Подп. и дата	Инв. № дубл.	Взам. инв. №	Подп. и дата						Лист
Изм.	Лист	№ докум.	Подпись	Дата	3271.102092.000 ПЗ					13

Недостатки данного подхода:

- неравномерное распределение продукции;
- необходимость в отслеживании остатков;
- риск образования ошибок человеческого фактора.

Динамическое адресное хранение – тип хранения, при котором за товаром не закрепляется конкретный адрес, а распределение по ячейкам регулирует либо персонал, либо учетная система. Особенность данного типа заключается в том, что продукция в системе закрепляется к определенной ячейке и сам товар располагается в ней. В случае отгрузки товара, необходимо списать данный товар из системы, отменив для него закрепление к ячейке, и освободить место в данной ячейке. Освободившееся место будет использовано в дальнейшем для другой продукции.

Преимущества данного подхода:

- равномерное распределение товаров, в соответствии с их размерами;
- возможность рационального использования складской площади;
- автоматизация бизнес-процессов.

Недостатки данного подхода:

- увеличение риска ошибок человеческого фактора;
- зависимость качества выполнения процессов от работоспособности системы.

Для реализации адресного хранения на складе необходимо модернизировать существующие бизнес-процессы. В данном случае будет происходить полная автоматизация работы склада, процессы которого состоят из:

- прием товара – получение продукции, проверка документов и их корректность;
- маркировка – добавление уникального кода на товар, данный процесс необходим для штрихкодирования товара в целях идентификации;
- перемещение в зону хранения – процесс транспортировки товара в строго отведенное для него место;

Инв. № подл.	Подп. и дата	Инв. № дубл.	Взам. инв. №	Подп. и дата						
Изм.	Лист	№ докум.	Подпись	Дата	3271.102092.000 ПЗ					Лист
										14



- инвентаризация – процесс определения и расчета фактического наличия товара с данными, хранящимися в системе;

- комплектация – сборка заказа. Благодаря адресному хранению каждый сотрудник обладает информацией о том, где какой товар находится, что прямо влияет на процесс сборки;

- отгрузка – адресная система автоматически составляет задание на отбор продукции из определенных ячеек.

Для реализации адресного хранения необходимо присваивание штрихкода месту размещения товаров, а также присваивание штрихкода каждому товару, для идентификации. Для создания штрихкодов необходимо специальное программное обеспечение, способное по переданным данным сгенерировать штрихкод требуемого формата, а также предоставить возможность для печати созданного штрихкода.

#### 1.1.4 Штрихкодирование в области делопроизводства

Технология штрихкодирования также развивается в области делопроизводства и документооборота. Идентификация документов при помощи технологии штрихкодирования с каждым днем становится все более распространенной практикой среди предпринимателей. Документы получают штрихкодовые обозначения, которые предоставляют возможность использования, передачи и получения информации из общей базы данных компании сотрудникам, работающим в разных подразделениях, организациях, что ускоряет процесс обработки документов и сводит к нулю образование ошибок человеческого фактора [4].

Организации внедряют данную систему в целях улучшения процессов документооборота за счет улучшений качества сервиса, так как повышается скорость обслуживания клиентов и минимизация ошибок человеческого фактора, которые возникают при ежедневной работе с документами, тем самым создавая риск их потери, или при возникновении ошибки в ходе получения копий

Инв. № подл.	Подп. и дата	Взам. инв. №	Подп. и дата	Инв. № дубл.	Взам. инв. №	Подп. и дата	Инв. № подл.	
Изм.	Лист	№ докум.	Подпись	Дата	3271.102092.000 ПЗ			Лист
								15

контрагентом. Происходит систематизация учета документов за счет возможностей программного обеспечения, а также снижение издержек, благодаря повышению оперативности выполнения операций, связанных с документооборотом.

Процесс штрихкодирования документов делится на несколько частей:

1) Выгрузка документа. Отправка контрагенту сопровождается поиску необходимого первичного документа в базе данных. Далее происходит отправка на печать документа, и в ходе печати присваивается уникальный штрихкод на документ.

2) Отправка. После добавления штрихкода, документ предоставляется заказчику в целях получения обратной связи.

3) Возврат документа. Адресату необходимо утвердить и подписать документ, по возможности отправить экземпляр обратно отправителю. Сотрудник, получивший подписанный экземпляр от контрагента, отмечает возврат документа в системе электронного архива. Сам процесс возврата копии в базу состоит из нескольких этапов: считывание сканером или ввод цифр вручную, ожидание уведомления от программы электронного архива о получении и загрузке документа.

4) Регистрация документов. Документ помечается специальной меткой, подразумевающей получение документа и возможность доступа к документу в любое время.

5) Сохранение. После процесса регистрации и ввода документа со штрихкодом в программу, документ отправляется и хранится в архиве то количество времени, сколько это необходимо.

Для реализации данного процесса необходимо специальное программное обеспечение, способное генерировать штрихкоды требуемого формата по имеющимся данным.

Инв. № подл.	Подп. и дата	Инв. № дубл.	Взам. инв. №	Подп. и дата	<div>3271.102092.000 ПЗ</div>	Лист
						16
Изм.	Лист	№ докум.	Подпись	Дата		

### 1.1.5 Штрихкодирование в обеспечении безопасности

Организации, занимающиеся предоставлением услуг по обеспечению безопасности, с каждым годом улучшают существующие системы, применяя штриховое кодирования в разнообразных направлениях. Использование штрихкода для идентификации личности становится повсеместной практикой в крупных организациях и компаниях. При помощи системы контроля доступа по штрихкоду регулируется и ограничивается доступ к производственным объектам для неподтвержденных сотрудников. Также данная система используется в ходе проверки посетителей и участников различных мероприятий. Участники данных мероприятий приобретают билеты участников, на которых распечатан сгенерированный код билета. При помощи специального устройства, контроллер считывает штрихкод на билете посетителя и на устройстве вывода отображается информация о владельце, которую сотрудник сравнивает с информацией на билете и выносит решение о подлинности данного билета, а также о пропуске данного посетителя на мероприятие [35].

Преимущества данного типа контроля доступа:

1) Недорогая система относительно других. Данная система славится доступностью для клиентов, так как для реализации системы штрихкодирования достаточно только специального оборудования и программного обеспечения. Персоналу нет необходимости в сдаче биометрических данных, так как достаточно персональных, что сокращает издержки временных и денежных ресурсов.

2) Обновление и перевыпуск штрихкода для сотрудника. Данная система предоставляет возможность обновления информации о сотруднике в случае смены персональных данных и перевыпуск штрихкода, исходя из обновленной информации.

3) Слежка за трафиком. Имеется контроль за персоналом и их передвижению по предприятию. Сотрудникам необходимо использовать штрихкод на специальных терминалах для захода в различные области предприятия. После

Инв. № подл.	Подп. и дата	Инв. № дубл.	Взам. инв. №	Подп. и дата						Лист
Изм.	Лист	№ докум.	Подпись	Дата	3271.102092.000 ПЗ					17

идентификации, данные о предоставлении доступа передаются сохраняются в базе данных предприятия, как при входе, так и выходе.

4) Разовое посещение. Имеется возможность предоставления доступа к предприятию для разового посещения. В данном случае используемые штрихкод используется только при единственном входе и выходе с предприятия, а в дальнейшем он становится недоступным для данного посетителя.

Недостатком данного типа контроля доступа является низкий уровень безопасности, так как возможность копирования штрихкода создает вероятность несанкционированного доступа на предприятие, что негативно влияет на уровень безопасности предприятия.

#### 1.1.6 ERP-система

ERP-система – программное обеспечение, представляющее собой комплекс решений по управлению компанией. Аббревиатура *ERP* расшифровывается как *enterprise resource planning*, что в переводе с английского означает – планирование ресурсов предприятия. Данная технология хранит данные бизнес-процессов компании и связывает имеющуюся информацию. Такие системы используются для оптимизирования и автоматизации деятельности отделов продаж, закупок, управление персоналом, бухгалтерии и многих других. По большей части использование данных систем свойственно компаниям не относящихся к малому бизнесу, как правило крупным организациям, так как имеется необходимость в обработке и передаче крупных объемов данных в короткие сроки. В ходе анализа было выявлено, что небольшие организации избегают использования данных систем в силу отсутствия необходимых ресурсов для поддержания системы в долгосрочной перспективе. В данный момент распространено использование модульных ERP-систем в соответствии с рисунком 2, так как подобная система предоставляет возможности использования конкретного программного обеспечения для определенных целей, а так как каждая корпорация требует

Инв. № подл.	Подп. и дата	Инв. № дубл.	Взам. инв. №	Подп. и дата	<div>3271.102092.000 ПЗ</div>	Лист
						18
						Изм. Лист № докум. Подпись Дата

выполнения определенного рода задач, то и требования к системе будут составляться исходя из существующих задач [5].



Рисунок 2 – Представление *ERP*-системы

Термин *ERP*-система используется для обозначения крупных интегрированных систем, которые обеспечивают контроль за осуществлением производства, продаж, закупок и учета в процессе выполнения клиентских заказов и многого другого.

До внедрения подобных систем, компании сталкиваются с рядом проблем, для решения которых необходимы затраты имеющихся временных и человеческих ресурсов [6].

Задачи, решаемые при помощи *ERP*-системы:

- на производстве распространено использование нескольких программных обеспечений для разных отделов. Часто передача данных из одной программы в другую затруднительна, либо невозможна, что является проблемой для большого количества предприятий. *ERP*-система предоставляет возможность использования

Инв. № подл.	Подп. и дата	Инв. № дубл.	Взам. инв. №	Подп. и дата
Изм.	Лист	№ докум.	Подпись	Дата
3271.102092.000 ПЗ				Лист
				19

и обработки данных, а также передачу информации из одного компонента системы в другую без затруднений;

- наличие того функционала программного обеспечения недостаточно для выполнения требуемых задач предприятия. Также процесс расширяемости организации замедляется в силу ограниченности в возможностях используемого программного обеспечения. Современные *ERP*-системы предоставляют доступ к большому количеству необходимых в работе инструментов;

- часто возникают ситуации, в которых персонал производства может совершить ряд ошибок, в силу своей низкой компетенции в определенных вопросах, поэтому необходимо снижение вероятности появления ошибок человеческого фактора. На данный момент функционала нынешних *ERP*-систем достаточно для сокращения данной вероятности.

Архитектура системы – это организация системы, представленная в виде компонентов, а также взаимоотношениях их друг с другом в общей среде.

Архитектура *ERP*-системы включает в себя несколько частей:

1) Платформа – специальная среда для использования встроенных функций, в дополнение к которым идет руководство по эксплуатации системы и функций.

2) Базовые функции – программное обеспечение, встроенное в систему для выполнения определенного спектра задач.

3) База данных – хранилище данных, расположенное на сервере вместе с инструментами обработки информации. Вся вносимая информация сохраняется здесь для дальнейшего использования в модулях системы.

4) Модуль – компонент, представляющий собой дополнительное программное обеспечение, цель использования которого, решение определенного спектра задач, которые не могут быть выполнены базовыми функциями системы.

По проведенному анализу существующих модулей систем выявились одни из самых распространенных сфер использования:

1) Бухгалтерия и финансы – модуль, ориентированный на проведение финансовых операций и задач, связанных с бухгалтерским делом. Модуль

Инв. № подл.	Подп. и дата	Инв. № дубл.	Взам. инв. №	Подп. и дата						Лист
Изм.	Лист	№ докум.	Подпись	Дата	3271.102092.000 ПЗ					20

позволяет генерировать и хранить финансовые документы, упрощает процесс создания платежных квитанций и налоговых отчетов.

2) Управление складом – обработка данных о запасах, местоположении товаров, сроках доставки, даты упаковки, отгрузки и многое другое. Предоставление возможности планирования передвижения товаров, трудозатрат и объем заказов у поставщиков.

3) Производство – обработка и передача информации о производстве в данный момент времени. При помощи подобных инструментов имеется возможность проверки, например, затраченного сырья, данные датчиков или загруженность объектов цеха. Также имеется возможность оптимизации и контроля процессов предприятия.

4) Контроль проекта – данный модуль предоставляет инструменты по управлению бюджета, отслеживанию и корректировке сроков, выставлению целей и задач и другие инструменты, ориентированные на проектный менеджмент.

5) Контроль персонала – предлагаемые программные модули предоставляют инструменты по управлению и контролю деятельности персонала и работников организации. В данном модуле имеются такие функции, как создание и редактирование кадрового учета, расчет заработной платы, учет времени, создание рабочего графика, редактирование личных карточек персонала и многие другие инструменты менеджера персонала.

6) Контроль поставок – данный программный модуль включает в себя группу инструментов, связанных с управлением и отслеживанием информации о поставках. В данном модуле имеются такие инструменты, как контроль закупок, анализ спроса, управление логистикой, редактирование контрагентов.

В ходе исследования и эксплуатации существующих систем был вынесен ряд недостатков *ERP*-систем, наличие которых имеет весомое влияние на мнение руководства организаций об интеграции системы в компанию. Список недостатков состоит из таких пунктов:

Инв. № подл.	Подп. и дата	Инв. № дубл.	Взам. инв. №	Подп. и дата	<div>3271.102092.000 ПЗ</div>	Лист
						21
						Изм.

1) Тяжесть в использовании уже существующих данных. На производстве бывают случаи негативного опыта использования *ERP*-системы. При переходе на новую *ERP* имеются трудности в передаче данных из старой системы на новую.

2) При переходе на новую систему необходимо предоставить техническое оборудование и настроить все технические платформы на пользование новой *ERP*-системой, что в свою очередь также влияет на затрату временных ресурсов.

3) Многие компании не имеют возможности для внедрения сложных программных комплексов, по причине отсутствия требуемых денежных ресурсов.

4) После внедрения новой системы, компании не берут в расчет затраты на обслуживание и поддержку системы в дальнейшем, что может сказаться на долговечность использования системы в компании.

5) Внедрение системы может занимать продолжительное количество времени, из-за того, что сотрудникам необходимо изучить новую систему, и получить опыт в использовании необходимых в работе инструментов.

*ERP*-системы по особенностям архитектуры разделяются на несколько видов. Необходимые особенности архитектуры подбираются в соответствии с требованиями организации к внедряемой системе. *ERP*-системы классифицируются по способу хранения информации, по виду организации, по интерфейсу. Существующие системы по способу хранения информации делятся на облачные и внутренние:

1) Облачные *ERP*-системы – это комплекс решений для управления ресурсами организации, которые проводят обработку данных на сервис-провайдере. Пользователи, используя данный вид систем, удаленно подключаются к специально разработанному интерфейсу системы для работы с ней. Большая часть компаний в силу отсутствия возможностей для использования собственных, независимых серверов в целях развертывания на них системы, стараются прибегать к приобретению доступа к системе у провайдера.

2) Внутренние *ERP*-системы – решения для управления ресурсами организации, которые проводят обработку данных на внутреннем сервере

Инв. № подл.	Подп. и дата	Взам. инв. №	Подп. и дата					
Инв. № дубл.								
Изм.	Лист	№ докум.	Подпись	Дата	3271.102092.000 ПЗ			
					Лист			
					22			



организации. Пользователи, используя данный вид систем, подключаются к встроенному интерфейсу системы.

*ERP*-системы делятся по типу интерфейса на стационарные программы и онлайн-сервисы.

1) Стационарные – программы, которые установлены на стационарные системы (компьютеры сотрудников). Особенность данных систем в возможности использования функционала без доступа в интернет, так как программное обеспечение функционирует с помощью внутренних средств стационарных систем.

2) Онлайн-сервис – сервисы, функционал которых, расположен на специализированных серверах. Возможность использования данной системы предоставляется только пользователям, которые зарегистрированы в компании, при помощи доступа в интернет [14].

Преимущества внедрения *ERP*-системы в компанию:

1) Централизованное управление. Руководители компаний имеют возможность контроля бизнес-процессов. Благодаря этому сокращается время обнаружения и исправления недочетов и недостатков в бизнес-планировании. Нисходящий поток данных проходит постоянную проверку, в целях обнаружения несовпадений и аномальных для бизнес-процессов явлений. Также предоставляются данные, рассчитанные на основе актуальных процессов в реальном времени, что делает их точными и корректными.

2) Эффективность. Автоматизация деятельности подразделений прямо влияет на производительность предприятия, за счет использования программных средств в деятельности каждого участника процесса. Персоналу нет необходимости в самостоятельном составлении расчетов, так как используются вычислительные мощности машины, как стационарной, так и удаленно расположенной. При помощи современных систем планирования имеется возможность определения продуктивности реальных процессов, в ходе расчета эффективности между действительными значениями и теоретическими. Сравнения

Инв. № подл.	Подп. и дата	Инв. № дубл.	Взам. инв. №	Подп. и дата	<div>3271.102092.000 ПЗ</div>	Лист
						23
						Изм.

предыдущих значений до внедрения изменений в процесс и после является важным пунктом определения результативности процесса в настоящее время.

3) Масштабируемость. Современные крупные организации стремятся к расширению своего бизнеса и занятию рынка в разных областях. Для выполнения подобных задач необходима надежная единая система управления, так как руководители компаний получают возможность для отслеживания процессов деятельности данных удаленных подразделений в реальном времени, что существенно влияет на контроль продуктивности и производительности, как сотрудников, так и отделов удаленных предприятий. Также благодаря интегрированным системам сотрудники имеют возможность использования, редактирования и отслеживания данных, полученных со всех предприятий в реальном времени.

4) Увеличение клиентской удовлетворенности. Качество выполнения задач, прямо зависит от качества реализованного функционала системы. Плохо функционирующая система может быть причиной сниженной производительности компании, что влияет на увеличение издержек, из-за повышения времени простоя процессов. Отлично функционирующая система предоставляет клиенту или пользователю максимально возможный функционал, что влияет на клиентскую удовлетворенность, так как пользователь сможет выполнять задачи в максимально возможные короткие сроки. Также нет необходимости в выполнении разного рода задач, так как они выполняются машиной и правильно структурированным программным обеспечением. Функционал современных *ERP*-систем позволяет, например, точно управлять заказами и запросами клиентов, а также своевременно обновлять данные для поддержки актуальности.

5) Уменьшение затрат. Снижение затрат – фактор, который влияет на общее мнение руководства организации о внедрении системы в свою компанию. Современные системы в ходе автоматизации и оптимизации бизнес-процессов предоставляют основания для сокращения издержек. Так как многие процессы становятся автоматизированными, отсутствует необходимость в поддержке

Инв. № подл.	Подп. и дата	Инв. № дубл.	Взам. инв. №	Подп. и дата	<div>3271.102092.000 ПЗ</div>	Лист
						24
Изм.	Лист	№ докум.	Подпись	Дата		

сторонних средств для выполнения данных процессов, что значительно экономит имеющиеся ресурсы компании. А также оптимизация бизнес-процессов приводит к увеличению прибыли, за счет предоставления оптимизированных стратегий решения задач.

В ходе внедрения системы в компанию, возникают частые ошибки, которое совершает руководство по разным причинам, например, неопытность в проведении внедрений информационных систем, отсутствие необходимых знаний для внедрения системы [27].

Список проблем при внедрении *ERP*:

1) Недооценка ресурсов. Перед проведением внедрения новой системы необходимо провести точную оценку требующихся и имеющихся для изменений ресурсов. Так как при возникновении ситуации полного отсутствия требуемых ресурсов, имеется вероятность полного отказа от внедрения, либо продолжительного застоя.

2) Ошибки планирования. Отсутствие тщательно продуманной стратегии внедрения может привести к принятию неверных решений в построении системы. Важно понимать то, что разработчики могут столкнуться с неопределенностью в приоритетах выполнения задач, что ведет к издержкам имеющихся ресурсов.

3) Неопределенность потребностей. Во время процесса внедрения, компании, в силу отсутствия необходимого опыта внедрения информационных систем, сталкиваются с проблемами в определении требуемого функционала для будущей системы.

Существующие *ERP* решения включают в себя как ряд, достоинств так и недостатков. Клиентам необходим полноценный функционал, выполняющий необходимые задачи при наименьшем использовании ресурсов.


Представленные области использования являются актуальными и развиваются в разных направлениях. Необходимость использования технологии штрихкодирования в каждой из них является обязательной частью большинства бизнес-процессов, без которой невозможно выполнение задач.

Инв. № подл.	Подп. и дата	Инв. № дубл.	Взам. инв. №	Подп. и дата						Лист
Изм.	Лист	№ докум.	Подпись	Дата	3271.102092.000 ПЗ					25

## 1.2 Виды штрихкодов

Различают несколько типов штрихкодов в соответствии с таблицей 1. Данный список собран исходя из анализа проблемных областей.

Таблица 1 – Типы штрихкодов

Тип штрихкода	Описание	Пример
<i>Code 128</i>	Используется для алфавитно-цифровых и цифровых штрихкодов. Можно закодировать все 128 <i>ASCII</i> символов.	 123456
<i>Codabar</i>	Штриховой код, позволяющий кодировать числа от 0 до 9, символы -, \$, :, /, ., + и четыре буквы ( <i>A, B, C, D</i> ).	 123456478
<i>Code 39</i>	Штрихкодированное обозначение переменной длин. Спецификация <i>Code 39</i> определяет 43 символа: заглавные латинские буквы (от <i>A</i> до <i>Z</i> ), цифры (от 0 до 9) и специальные символы (-, ., \$, /, +, %, и пробел).	 123456
<i>EAN-2</i>	Дополнение к штрихкоду <i>EAN-13</i> . Часто используется в газетной индустрии.	 11
<i>EAN-5</i>	Дополнение к штрихкоду <i>EAN-13</i> . Часто используется в книжной продукции, в частности, как код цена для <i>ISBN</i> .	 11212
<i>UPC</i>	Символика штрихкодов, используемая для отслеживания товаров в магазинах	 1 23456 78910 4

Инв. № подл.	Подп. и дата
Инв. № дубл.	Взам. инв. №
Подп. и дата	Подп. и дата

Продолжение таблицы 1

<i>EAN-8</i>	Штрихкод <i>EAN-8</i> является дополнением к более длинному коду <i>EAN-13</i> . Он предназначен для использования на маленьких упаковках	
<i>EAN-13</i>	Графическое изображение уникального международного товарного номера в машиночитаемом виде. Товар, идентичный по наименованию и характеристикам, но имеющий различную упаковку (граммы, литры), маркируется разными номерами в штрих-коде.	
<i>ITF-14</i>	Реализация штрихкода <i>interleaved 2 of 5</i> для кодирования <i>Global Trade Item Number</i> . <i>ITF-14</i> всегда кодирует 14 цифр.	
<i>MSI</i>	Непрерывный код, различной длины. В ходе отображения используются цифры от 0 до 9. Для контроля могут использоваться одна или две контрольных цифры.	
<i>Pharmacode</i>	Стандарт штрихового кода, который широко используется в фармацевтической деятельности и промышленности	
<i>Data Matrix</i>	Двумерный матричный штрихкод, содержащий черные и белые ячейки, упорядоченные в квадратный или прямоугольный массив. Может быть закодирован как текст, так и произвольные данные.	
<i>QR-код</i>	Тип матричных штриховых кодов (или двухмерных штриховых кодов). <i>QR-код</i> состоит из чёрных квадратов, расположенных в квадратной сетке на белом фоне, которые могут считываться с помощью устройств обработки изображений.	

Приведенная таблица иллюстрирует, используемые типы штрихкодов, необходимые в реализации программного обеспечения, а именно модуля генерации штрихкодов.

Инв. № подл.	Подп. и дата
Инв. № дубл.	Взам. инв. №
Подп. и дата	

Изм.	Лист	№ докум.	Подпись	Дата	3271.102092.000 ПЗ	Лист
						27

### 1.3 Цель и задачи выпускной квалификационной работы

Основной целью выпускной квалификационной работы является повышение эффективности складского учета при помощи модуля генерации штрихкодов для *ERP*-системы. Практическая ценность ВКР определяется тем, в какой мере содержащиеся в ней предложения и рекомендации способствуют улучшению деятельности предприятия, могут быть применены и положительно оценены его руководством.

Для реализации поставленной цели необходимо выполнение следующих задач:

- проведение анализа предметной области;
- проведение анализа предлагаемого бизнес-процесса;
- проектирование моделей предлагаемого бизнес-процесса;
- разработка моделей базы данных;
- проектирование модели архитектуры системы;
- создание модуля для генерации штрихкодов и их печати.

Корректно сформулированные цели и задачи создания системы задают ход для успешной реализации и разработки программного продукта. Цель необходима для структурирования и управления ходом выполнения проекта, который состоит из поочередного выполнения поставленных задач.

Инв. № подл.	Подп. и дата	Инв. № дубл.	Взам. инв. №	Подп. и дата						
Изм.	Лист	№ докум.	Подпись	Дата	3271.102092.000 ПЗ					Лист
										28

## 2 Проектирование моделей бизнес-процесса

### 2.1 Требования к программному продукту

Модуль генерации штрихкодов для *ERP*-системы должен выполнять ряд следующих функций:

- генерация штрихкодов по передаваемым данным;
- печать сгенерированных штрихкодов;
- печать переданных штрихкодов;
- разработанный интерфейс модуля генерации;
- способы обращения к заявленным функциям.

Интерфейс должен содержать главную страницу: «Генерация штрихкода». На данной странице должен отображаться интуитивно понятный интерфейс для пользователя [16].

Для достижения данной цели используется несколько объектов:

- блок ввода;
- блок выбора типа штрихкода;
- кнопка для генерации;
- кнопка для отправки на печать.

Блок ввода необходим пользователю для ввода имеющихся данных и передачи их для обработки в базу данных системы. Блок выбора типа штрихкода необходим для предоставления пользователю выбора типа штрихкода на генерацию. Кнопка для генерации необходима для предоставления пользователю возможности генерации штрихкода и отправки на базу данных введенной информации. Кнопка отправки на печать необходима для открытия специального окна браузера для редактирования электронного шаблона для печати и последующей отправки на печать в термопринтер или принтер.

Инв. № подл.	Подп. и дата	Взам. инв. №	Подп. и дата

Изм.	Лист	№ докум.	Подпись	Дата	3271.102092.000 ПЗ	Лист
						29

## 2.2 Выбор методологии моделирования бизнес-процесса

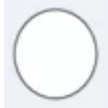


### 2.2.1 Описание методологии BPMN

Для реализации модели исследуемого бизнес-процесса, будет использоваться методология BPMN (*Business Process Model and Notation*).

Данный язык моделирования бизнес-процессов является промежуточным местом между визуализацией и воплощением бизнес-процесса. С помощью моделирования описываются бизнес-процессы любой сложности для различных систем управления [7].

Для отображения используются специальные графические символы в соответствии с таблицей 2.

Таблица 2 – Графические обозначения методологии BPMN

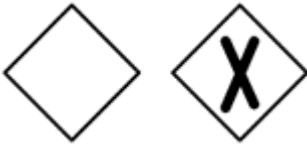


Объект	Описание	Графическое отображение
Событие ( <i>Event</i> )	Используется для обозначения ситуаций, возникающих в ходу выполнения процесса.	Начальное событие:  Конечное событие: 
Действие ( <i>Activity</i> )	Задача, которую необходимо выполнить на соответствующем этапе бизнес-процесс. В моделировании процессов выделяются два основных вида действий: подпроцесс и задача.	Абстрактная задача – необходим в случае обозначения операции, не имеющей дальнейшей декомпозиции. 

Инв. № подл.	Подп. и дата
Инв. № дубл.	Взам. инв. №
Подп. и дата	
Инв. № подл.	

Изм.	Лист	№ докум.	Подпись	Дата	3271.102092.000 ПЗ	Лист
						30



Продолжение таблицы 2

Шлюз ( <i>GATEWAY</i> )	Управляющий узел, необходимый при условном разветвлении бизнес-процесса.	Исключающий или – используется в случае отображения альтернативных или сходящихся потоков.    Оператор «И» – используется для ветвления, превращая поток управления в множество параллельных.    Оператор «ИЛИ» – активируется одна или более выходящих ветвей, при использовании ветвления.
Объекты данных ( <i>DATA OBJECT</i> )	Элемент, изображающий какие данные необходимы действиям для выполнения или какие данные определенные действия производят.	
Поток	Стрелки, изображающие движение по процессам и порядок их выполнения.	Поток управления – изображение порядка выполнения процесса.    Ассоциация – изображает связь объектов данных и баз данных с процессами.  

Инв. № подл.	Подп. и дата	Инв. № дубл.	Взам. инв. №	Подп. и дата

Изм.	Лист	№ докум.	Подпись	Дата

3271.102092.000 ПЗ

Преимущества данного языка моделирования:

1) Доступное отображение процесса. Данный язык моделирования позволяет описать процесс таким образом, что он становится понятным и доступным для восприятия всем участникам, как внутри, так и снаружи процесса.

2) Многофункциональность. *BPMN* можно использовать для описания процессов различных областей, поэтому он славится своей универсальностью. Бизнес-аналитики используют его как в фирмах малого бизнеса, так и в крупных компаниях [29].

Недостатки данного языка моделирования:

1) Обильное наличие терминов. Необходимо ознакомиться с терминами и понятиями, присущими данному языку моделирования, для корректного моделирования процессов.

2) Знание бизнес-процесса. Детальное знание бизнес-процесса необходимо для выстраивания грамотной структуры модели процесса. Необходимо отобразить последовательность действий каждого сотрудника на модели [17].

Наглядная схема в данной методологии предоставляет информацию о ходе процесса. Процесс изображается в виде последовательности рабочих действий и перемещении потоков информации.

## 2.2.2 Разработка модели функционирования бизнес-процесса

Бизнес-процесс реализации модуля генерации штрихкодов был описан с использованием нотации *BPMN*. Данная модель описывает бизнес-процесс приемки и штрихкодирования товара, который включает в себя процесс генерации и штрихкодирования товара. Модель представлена на рисунке 4.

Этап штрихкодирования товара состоит из коммуникации сотрудника склада и поставщика, а также работы системы создания и выдачи штрихкода.

В ходе процесса передачи товара, поставщик передает товар вместе с накладной — документом, фиксирующим передачу товарно-материальных ценностей от одного лица другому.

Подп. и дата		Взам. инв. №		Инв. № дубл.		Подп. и дата		Инв. № подл.	
Изм.	Лист	№ докум.	Подпись	Дата	3271.102092.000 ПЗ				Лист
									32

Данные о полученном товаре, необходимо внести в базу данных *ERP*-системы. Сотрудник склада вносит все имеющиеся данные о товаре и при необходимости редактирует. Все внесенные позиции передаются в базу данных *ERP*.

При условии существования введенного товара в базе данных, происходит отправка, ранее сгенерированного штрихкода данного товара, из базы данных на компьютер сотрудника. В случае отсутствия данного товара в базе данных системы, происходит генерация нового штрихкода для данной позиции, который вместе с информацией о товаре отправляется и сохраняется в базу данных *ERP* [8].

Полученный ранее штрихкод отправляется на принтер для печати. Этикетка с напечатанным на ней штрихкодом прикрепляется к товару сотрудником склада.

Данный этап также представлен в нотации *BPMN* на рисунке 4.

Далее на мнемосхеме отображен этап продажи товара, который начинается с передачи товара покупателем продавцу. Все имеющиеся товары сканируются при помощи специального сканера штрихкода. На экране компьютера выдается считанный товар, с данными, соответствующими данной позиции.

Процесс формирования заказа происходит из считанных товаров и их данных. Продавец обязан проверить сформированный заказ и, в случае обнаружения ошибки, исправить ее в кратчайший срок.

Покупатель, получая информацию о цене заказа, должен передать оплату продавцу.

Внесенная оплата переходит в обработку и сохраняется в базе данных *ERP*-системы.

Информация о продаже товаров фиксируется в базе данных. Происходит составление и формирование электронного чека, а также запрос на принтер для распечатывания бумажного чека. После получения чека продавец передает его покупателю вместе с купленным товаром.

Мнемосхема и *BPMN* модель процессов отображены на рисунках 3 и 4 соответственно:

Инв. № подл.	Подп. и дата	Инв. № дубл.	Взам. инв. №	Подп. и дата						Лист
Изм.	Лист	№ докум.	Подпись	Дата	3271.102092.000 ПЗ					33

Инв. № подл.	Подп. и дата	Инв. № дубл.	Взам. инв. №	Подп. и дата

Изм.	Лист	№ докум.	Подпись	Дата

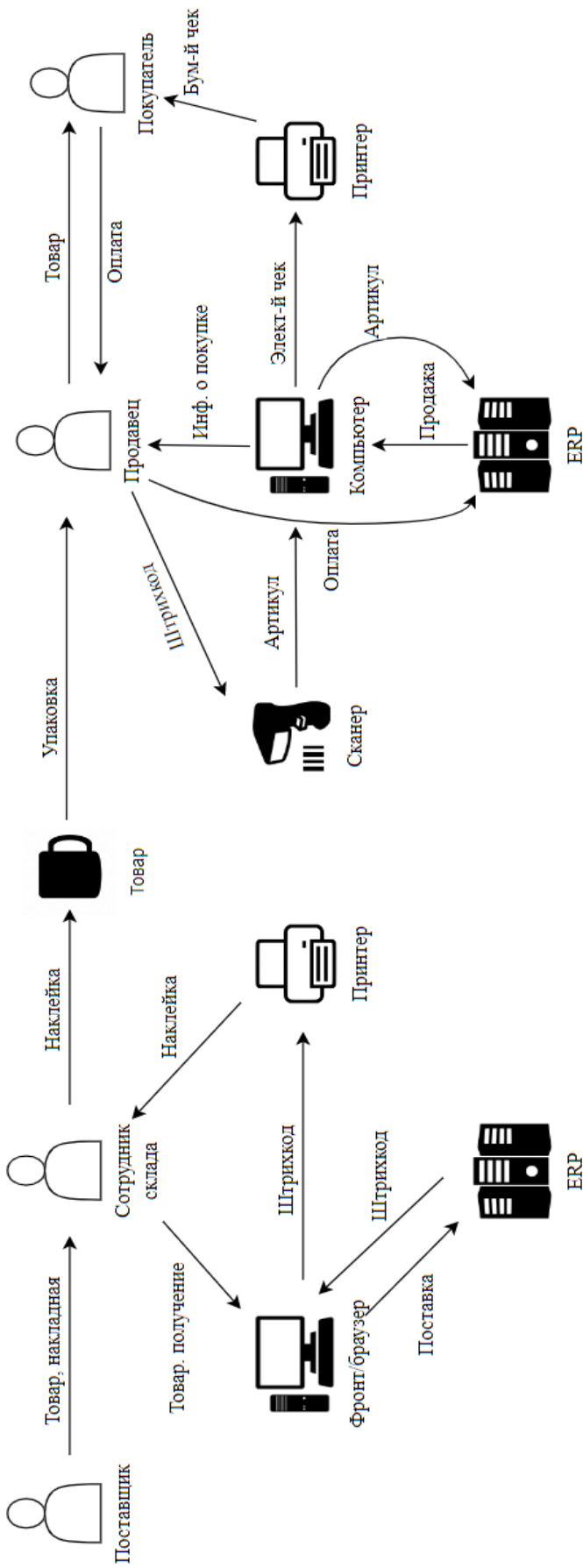


Рисунок 3 – Процесс штрихкодирования товара и передачи его покупателю



Инв. № подл.	Подп. и дата	Инв. № дубл.	Взам. инв. №	Подп. и дата

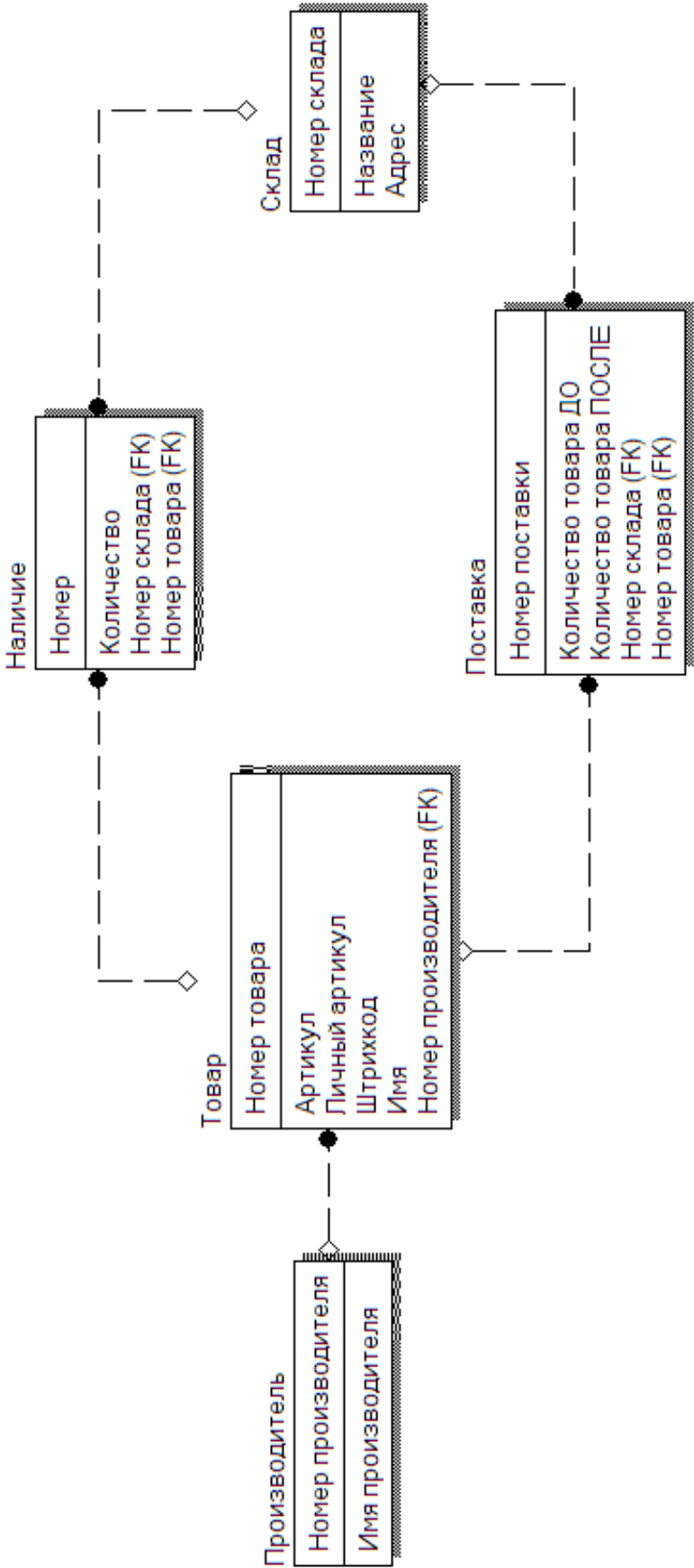


Рисунок 5 – Логическая модель базы данных

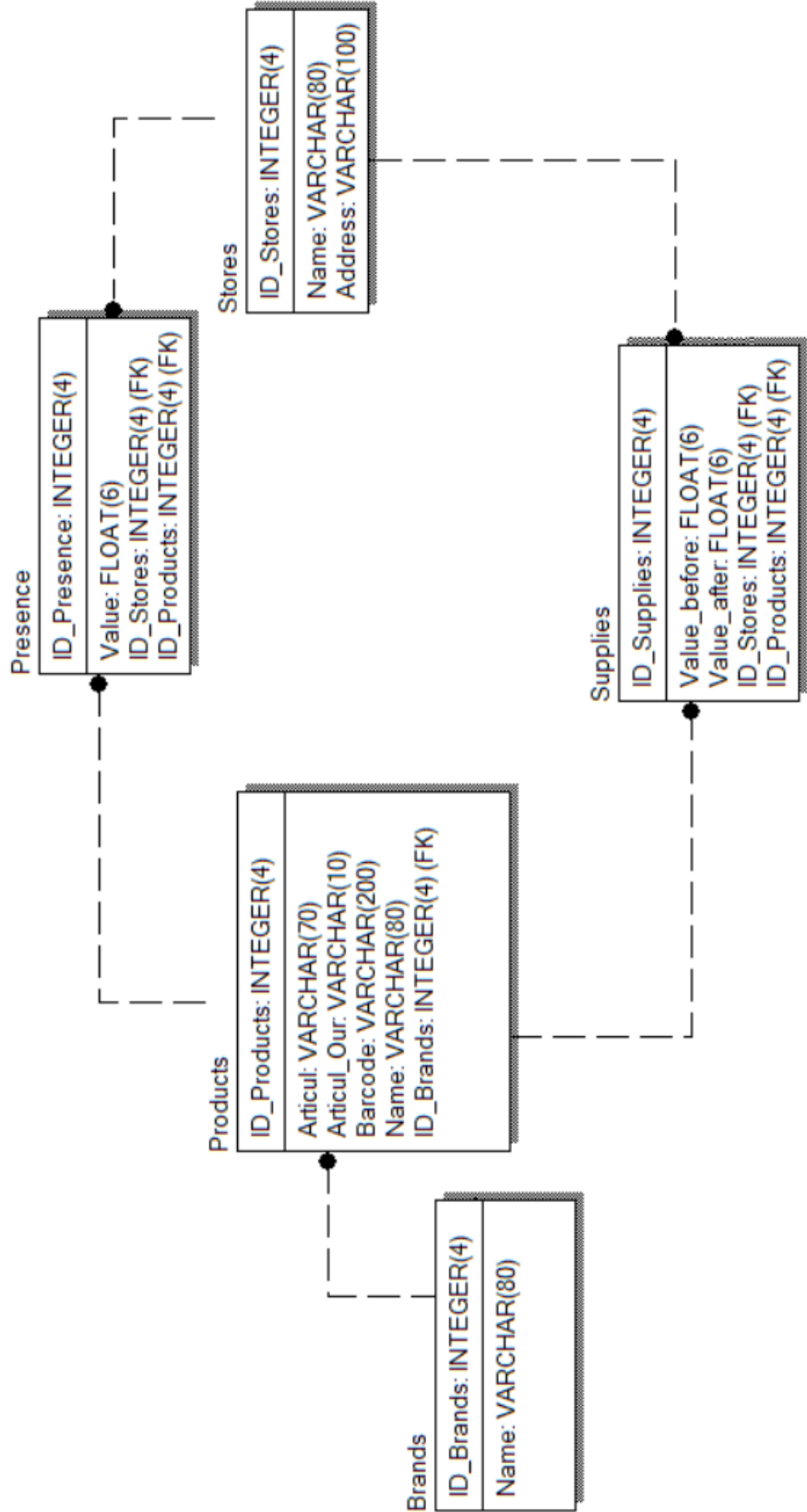


Рисунок 6 – Физическая модель базы данных



В таблице 3 хранится информация о товаре. В нем содержится информация об артикуле производителя, собственном артикуле, штрихкод, имя, производитель. Также необходимо отметить присвоение идентификационного номера для каждого товара. Артикулы от производителей могут совпадать и создавать некорректные ситуации в системе учета. Для правильной работы системы учета, необходимо выдавать товару артикул от организации в процессе приемки. Сгенерированный штрихкод будет находится в таблице, предоставляя полноценный свод информации о конкретном товаре.

Таблица 3 – *Products*

Название столбца	Тип	Описание	Ключ	Длина	Значение по умолчанию	Автоинкрементация
<i>ID</i>	<i>Int</i>	Главный ключ	<i>Pr.key</i>	4	Инкрементация +1	1
<i>Articul</i>	<i>Varchar (size)</i>	Артикул	-	70	<i>Null</i>	-
<i>Articul Our</i>	<i>Varchar (size)</i>	Собственный артикул	-	10	<i>Null</i>	-
<i>Barcode</i>	<i>Varchar (size)</i>	Штрихкод <i>json</i>	-	200	-	-
<i>Name</i>	<i>Varchar (size)</i>	Имя	-	80	-	-
<i>Brands</i>	<i>int</i>	Производитель	<i>For. key</i>	4	-	-

В таблице 4 хранится информация о компании производителе товара. В данной ней содержится информация названия компании производителя, а также идентификационный номер компании.

Таблица 4 – *Brands*

Название столбца	Тип	Описание	Ключ	Длина	Значение по умолчанию	Автоинкрементация
<i>ID</i>	<i>Int</i>	Главный ключ	<i>Pr.key</i>	4	Инкрементация +1	1
<i>Name</i>	<i>Varchar(size)</i>	Имя	-	80	-	-

Инв. № подл.	Подп. и дата
Инв. № дубл.	Взам. инв. №
Подп. и дата	

Изм.	Лист	№ докум.	Подпись	Дата	3271.102092.000 ПЗ	Лист
						39

В таблице 5 содержатся данные о складе. Данная информация необходима для отслеживания партии товара и контроле передвижения. Каждому существующему складу присвоен уникальный порядковый номер.

Таблица 5 – *Stores*

Название столбца	Тип	Описание	Ключ	Длина	Значение по умолчанию	Автоинкрементация
<i>ID</i>	<i>Int</i>	Главный ключ	<i>Pr.key</i>	4	Инкрементация +1	1
<i>Name</i>	<i>Varchar (size)</i>	Имя	-	80	-	-
<i>Address</i>	<i>Varchar (size)</i>	Адрес	-	100	-	-

В таблице 6 хранится информация о поставках определенного товара. Отмечаются такие особенности, как: идентификационный номер поставки, количество товара до поставки, количество товара после поставки, склад размещения и товар. Данные до поставки и после поставки необходимы для определения количества полученного товара.

Таблица 6 – *Supplies*

Название столбца	Тип	Описание	Ключ	Длина	Значение по умолчанию	Автоинкрементация
<i>ID</i>	<i>Int</i>	Главный ключ	<i>Pr.key</i>	4	Инкрементация +1	1
<i>Value before</i>	<i>Float</i>	Количество до поставки	-	6	-	-
<i>Value after</i>	<i>Float</i>	Количество после поставки	-	6	-	-
<i>Stores</i>	<i>Int</i>	Склад	<i>For.key</i>	4	-	-
<i>Products</i>	<i>Int</i>	Товар	<i>For.key</i>	4	-	-

Инв. № подл.	Подп. и дата
Инв. № дубл.	Взам. инв. №
Подп. и дата	
Инв. № подл.	

В таблице 7 хранится информация о наличии товара. Выделяются такие характеристики, как: идентификационный номер, количество товара, данные о складе, данные о товаре.

Таблица 7 – *Presence*

Название столбца	Тип	Описание	Ключ	Длина	Значение по умолчанию	Автоинкрементация
<i>ID</i>	<i>Int</i>	Главный ключ	<i>Pr.key</i>	4	Инкрементация +1	1
<i>Value</i>	<i>Float</i>	Количество	-	6	-	-
<i>Stores</i>	<i>Int</i>	Склад	<i>For.key</i>	4	-	-
<i>Products</i>	<i>Int</i>	Товар	<i>For.key</i>	4	-	-

Описание базы данных иллюстрирует строение, взаимосвязь объектов между собой и их зависимости. В данном случае наличие базы данных необходимо в некоторых случаях:

- хранение информации;
- получение и передача информации. Взаимодействие разных модулей системы подкрепляется использованием общей базы данных [31].

Для полноценного выполнения процессов необходимо выполнение данных пунктов, так как работа с информацией является одним из важных составляющих каждой из изучаемых областей и анализируемого процесса.

Инв. № подл.	Подп. и дата
Инв. № дубл.	Взам. инв. №
Подп. и дата	Инв. № подл.
Инв. № подл.	Подп. и дата

Изм.	Лист	№ докум.	Подпись	Дата	3271.102092.000 ПЗ	Лист
						41

## 2.4 Алгоритм работы программы

Важной частью данного программного обеспечения является алгоритм его работы.

Алгоритм генерации штрихкода состоит из нескольких этапов.

1) Получение входных данных. Для корректной работы алгоритма необходимы входные данные, в частности используется идентификационный номер, но также возможна передача различных типов данных, в зависимости от области использования генератора.

2) Выбор типа штрихкода. Предоставляется несколько типов штрихкодов. От выбора типа штрихкода зависит формат сгенерированного штрихкода.

3) Генерация штрихкода. После получения корректного набора данных, программный код выполняет генерацию штрихкода, строго в соответствии с особенностями выбранного типа штрихкода и введенных данных. На странице появляется необходимый штрихкод сразу после нажатия на кнопку: «Сгенерировать».

4) Проверка введенных данных. Для генерации определенного штрихкода необходим определенный набор данных. После нажатия на кнопку «Сгенерировать» программный код выполняет необходимые команды для проверки соответствия передаваемых данных к требованиям к генерации штрихкода. В случае обнаружения несоответствия перед клиентом выдается ошибка и способ ее исправления. После исправления необходимо нажать кнопку повторно.

5) Печать штрихкода. После создания штрихкода и его демонстрации на экране у клиента появляется возможность отправки штрихкода на печать. Для передачи на печать достаточно сгенерировать штрихкод и нажать на кнопку: «Печать».

Данный порядок действия корректно отображает особенности работы системы и ее специфику.

Инв. № подл.	Подп. и дата
Инв. № дубл.	Взам. инв. №
Подп. и дата	Инв. № подл.
Инв. № подл.	Подп. и дата

Изм.	Лист	№ докум.	Подпись	Дата	3271.102092.000 ПЗ	Лист
						42

## 2.5 Архитектура системы

Для реализации различных процессов в *ERP*-системе модульного типа, существуют различные группы модулей, например, модули учета товаров, модуль делопроизводства и другие. Разрабатываемый модуль генерации штрихкодов необходим для различных групп модулей, так как процессы соответствующих областей требует использования технологии штрихкодирования. Архитектура системы разработана в соответствии с рисунком 7.

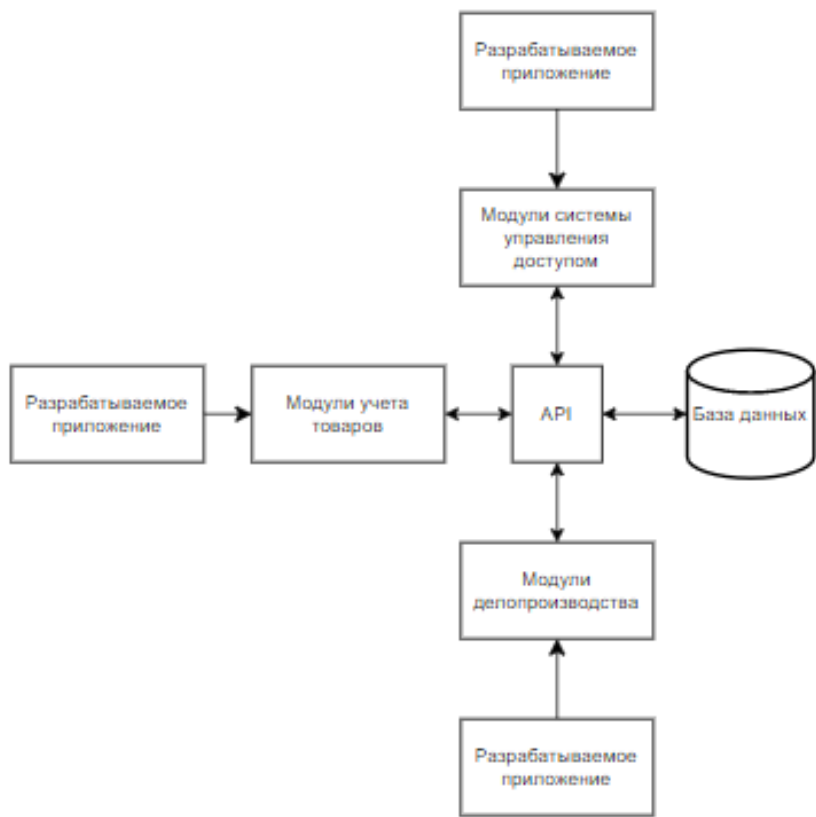


Рисунок 7 – Архитектура системы

Разрабатываемый модуль подключается и редактируется разработчиками под особенности данной группы модулей. Группы модулей подключаются через единый протокол *API* к базе данных для использования хранящейся там информации и для возможности выполнения различных действий над ней.

Инв. № подл.	Подп. и дата
Инв. № дубл.	Взам. инв. №
Подп. и дата	Подп. и дата

### 3 Разработка модуля генерации штрихкодов для *ERP*-системы

### 3.1 Используемые технологии

### 3.1.1 REST API

При переходе на сайт, браузеру необходимо отправить запрос на данные, которые будут отображаться перед пользователем. Для эффективной и оптимизированной передачи данных был разработан архитектурный подход *REST API*. Перед описанием понятия *REST API*, необходимо познакомиться с определением выражения *API* [23].

*API* (Программный интерфейс приложения) – это набор инструментов, необходимый для работы одних программ вместе с другими. Возможности данной технологии предоставляют работу программ друг с другом даже на разных вычислительных машинах. Пример: использование банковских сервисов оплаты при заказе товара в интернет-магазине. В общении между программами существуют спецификации и сообщения. Спецификации – описание правил для программного интерфейса, а сообщения – порции данных, которыми обмениваются приложения.

Для внедрения в систему программного интерфейса приложений существует ряд причин:

- необходимость в оптимизации обмена ресурсов и данных;
- необходимость в проверке и отслеживании доступов по аутентификации и распределении прав доступа;
- требуется возможность в подключении сторонних приложений;
- требуется возможность в обеспечении безопасности передаваемых и получаемых данных;
- разработка микросервисных серверных приложений.

*REST API* – это архитектурный подход, устанавливающий ограничения для *API*. Технология *REST* предоставляет возможности поддержки и обеспечения связи в необходимых масштабах.

Инв. № подл.	Подп. и дата	Инв. № дубл.	Взам. инв. №	Подп. и дата

программного интерфейса, а сообщения – порции данных, которыми обмениваются приложения.

Для внедрения в систему программного интерфейса приложений существует ряд причин:

- необходимость в оптимизации обмена ресурсов и данных;
- необходимость в проверке и отслеживании доступов по аутентификации и распределении прав доступа;
- требуется возможность в подключении сторонних приложений;
- требуется возможность в обеспечении безопасности передаваемых и получаемых данных;
- разработка микросервисных серверных приложений.

*REST API* – это архитектурный подход, устанавливающий ограничения для *API*. Технология *REST* предоставляет возможности поддержки и обеспечения связи в необходимых масштабах.

Инв. № подл.	Подп. и дата	Инв. № дубл.	Взам. инв. №	Подп. и дата

Изм.	Лист	№ докум.	Подпись	Дата

**3271.102092.000 ПЗ**

Лист 44

### Преимущества *REST API*:

- системы, использующие *REST API*, эффективно масштабируются благодаря оптимизации взаимодействия между сервером и клиентом по *REST*;
- поддержка разделение клиента и сервера, которая разделяет различные серверные компоненты для независимого развития. Смена платформы серверного приложения не влияет на клиентское приложение. Происходит повышение гибкости;
- *REST API* может использоваться в разного рода технологиях. Возможно создание как клиентских, так и серверных приложений на разных языках программирования, не влияя на структуру *API*;
- *HTTP*-протокол реализован для всех языков программирования, во всех операционных системах, в отличие от других протоколов, что дает предоставляет широкий спектр возможностей использования.

Основными компонентами в работе *REST API* являются: клиент, сервер, источник.

Клиент – программное обеспечение, которое запрашивает у сервера доступ к ресурсам.

Сервер – программа, в которой хранятся и обрабатываются ресурсы. Данное программное обеспечение предоставляет доступ клиенту при помощи *API* (программного интерфейса приложения) к данным и функционалу.

Ресурс – данные, участвующие в обмене информации между клиентом и сервером (например, документы, изображения, временный сервис). Представлением ресурса чаще всего именуют состояние ресурса, в нем содержатся данные, метаданные, гипермедиа ссылок [9].

Принцип работы происходит в соответствии с рисунком 8.

Инв. № подл.	Подп. и дата	Инв. № дубл.	Взам. инв. №	Подп. и дата						
Изм.	Лист	№ докум.	Подпись	Дата	3271.102092.000 ПЗ					Лист
										45



Рисунок 8 – Принцип работы *REST API*

Происходит отправка запроса от клиента на сервер. При помощи метода *HTTP (HyperText Transfer Protocol)* происходит передача серверу информации о действии, которое необходимо провести с запрашиваемым ресурсом. Для того чтобы выполнялись некоторые действия с запрашиваемым ресурсом используются методы:

- *GET* используется при чтении информации. Данный метод необходим для возврата данных при запросе с сервера. Данный метод особенен тем, что возвращаемая информация не меняется и не удаляется, а только возвращается для чтения;

- *POST* используется для создания нового объекта или для изменения;

- *PUT* используется для редактирования существующего объекта. Часто применяется в ходе обновления данных. В отличие от *POST*, повторные отправки запроса выдают один и тот же результат в *RESTful*;

- *DELETE* используется для удаления ресурса с сервера.

Данные операции представляют собой действия *CRUD*.

Для ответов и запросов существуют собственные структуры и элементы, им свойственные. Как правило запросы состоят из следующих элементов:

- 1) тело запроса – это информация, переданная браузеру при запросе страницы;

Инв. № подл.	Подп. и дата
Инв. № дубл.	Взам. инв. №
Подп. и дата	Подп. и дата

Изм.	Лист	№ докум.	Подпись	Дата	3271.102092.000 ПЗ	Лист
						46



2) конечная точка – адрес отправки запроса;

3) параметры пути, определяющие *URL*;

4) параметры запросы, для определения необходимой информации о ресурсе;

5) параметры *cookie*, для аутентификации клиентов;

6) заголовки – данные, необходимые для корректной обработки запроса, так как в них присутствуют *HTTP* заголовки. В основном состоящие из формата передаваемых данных, их особенностях, информации о протоколе обмена и многих других.

Структура ответов также имеет собственные отличительные элементы:

1) заголовки, которые передают данные для корректной обработки ответа;

2) тело ответа включает в себя информацию, которую запрашивал клиент во ходе отправки запроса;

3) код ответа, обозначающее собой выполненный запрос.

Для создания *REST* интерфейса необходимо соблюдать ряд принципов, которые регулируют правила отправки запросов и ответов.

1) Архитектура *Client-server*. Необходимо использовать архитектуру *client-server*. Ее особенность в том, что клиент требует ресурсы, не имея связи с хранилищем данных. Сервер занимается хранением данных, обработкой входящих запросов, отправкой ответов клиенту, но при этом не имея связи с пользовательским интерфейсом.

2) Унифицированный интерфейс представляет собой использование единого способа работы с сервером. Данная концепция подразумевает под собой реализацию идентичных *API* запросов к ресурсу, несмотря на их точку происхождения. Существуют несколько принципов, архитектурных ограничений для реализации унифицированного интерфейса:

- клиенту предоставляется вся необходимая информация о связанных ресурсах для выполнения задач;

Инв. № подл.	Подп. и дата	Инв. № дубл.	Взам. инв. №	Подп. и дата	<div>3271.102092.000 ПЗ</div>	Лист
						47
Изм.	Лист	№ докум.	Подпись	Дата		

- у клиентов имеется представление о ресурсе для возможности редактирования его данных или их удаления. Данный принцип реализуется при помощи отправки метаданных сервером;

- каждый запрос должен идентифицировать ресурсы;

- каждый клиент получает всю имеющуюся информацию об обработке представлений. Данный принцип реализуется при помощи отправки метаданных сервером;

- на сервере не содержится информация о клиенте. Вся необходимая информация приходит с запросом;

- использование многоуровневой системы. Данный принцип подразумевает использование промежуточных узлов между сервером и клиентом, называемые прокси-сервер. Благодаря им происходит распределение нагрузки, а также регулировка хода запроса, в случае использования нескольких серверов.

Данный ряд принципов необходим для корректной работы системы по принципу *REST*.

### 3.1.2 Модель *OSI*

Сетевая модель *OSI* – сетевая модель стека сетевых протоколов *OSI/ISO*. При помощи данной модели устройства имеют возможность взаимодействовать друг с другом. Данная модель состоит из 7 уровней: прикладной, представления, сеансовый, транспортный, сетевой, канальный, физический. Положение прикладного уровня изображено на рисунке 9.

Инв. № подл.	Подп. и дата	Инв. № дубл.	Взам. инв. №	Подп. и дата					
Изм.	Лист	№ докум.	Подпись	Дата	3271.102092.000 ПЗ				Лист
									48

Семиуровневая модель OSI	
7	Прикладной уровень (application layer)
6	Уровень представления (presentation layer)
5	Сеансовый уровень (session layer)
4	Транспортный уровень (transport layer)
3	Сетевой уровень (network layer)
2	Канальный уровень (data link layer)
1	Физический уровень (physical layer)

Рисунок 9 – Уровни модели OSI

Седьмой уровень, или прикладной, обеспечивает взаимодействие сети и пользователя. На данном уровне выдается разрешение приложениям пользователя на получение доступа к сетевым службам, таким как обработчик запросов к базам данных, доступ к файлам и многим другим. Передача ошибок, служебной информации и формирование запросов к уровню представления [10].

Функции: сетевой виртуальный терминал, доступ к передаче файлов и управление ими.

Сетевой виртуальный терминал – программное обеспечение, которое предоставляет пользователю удаленный доступ к серверу при помощи программной эмуляции.

*FTAM (File Transfer Access and Management)* – протокол передачи данных, предоставляющий возможности распространения файлов при помощи сети и манипулирования ими. В отличие от других распространенных протоколов данных, *FTAM* открывает возможности печати и буферизации, создания сетевых файловых систем, поиск записей в удаленных базах данных.

Службы: почтовые службы, службы каталогов.

Программное обеспечение почтовой службы выполняет функцию организации пересылки и хранения электронной почты.

Инв. № подл.	Подп. и дата
Инв. № дубл.	Взам. инв. №
Подп. и дата	Подп. и дата
Инв. № подл.	Инв. № дубл.

Изм.	Лист	№ докум.	Подпись	Дата	3271.102092.000 ПЗ	Лист
						49

Программное обеспечение службы каталогов представляет собой инфраструктуру, участвующую в поисках и организации управления сетевыми ресурсами.

### 3.1.3 *JavaScript, HTML, CSS*

Для разработки нашего проекта мы используем языки программирования *JavaScript, HTML, CSS*.

Языки верстки *HTML, CSS* являются наилучшими технологиями для разработки веб-интерфейсов и веб-приложений. *HTML (Hypertext Markup Language)* – это код, который используется для структурирования и отображения веб-страницы и её контента. Веб-браузеры получают *HTML* документ от сервера по протоколам *HTTP/HTTPS* или открывают с локального диска, далее интерпретируют код в интерфейс, который будет отображаться на экране монитора.

*CSS* — каскадные таблицы стилей для придания внешнего вида веб-странице. Данный язык используется фронтенд разработчиками для придания оболочки блокам *HTML*-документа. Основной целью разработки *CSS* является ограждение и отделение описания логической структуры веб-страницы (которое производится с помощью *HTML* или других языков разметки) от описания внешнего вида этой веб-страницы (которое теперь производится с помощью формального языка *CSS*). Благодаря разграничению уменьшается нагрузка в содержимом *HTML*-документа, а также предоставляется больше возможностей в управлении содержимым документа [11]

*JavaScript* — это объектно-ориентированный язык программирования, широко используемый для создания динамических веб-страниц. Также поддерживает функциональные стили. Особенность в том, что *JavaScript* является скриптовым языком для написания сценариев веб-страниц. Это значит, что тот код, который вы напишите, будет выполняться построчно, от инструкции к инструкции. Сравнительный анализ данного языка с языком *Java* представлен в таблице 8.

Инв. № подл.	Подп. и дата	Взам. инв. №	Подп. и дата	Инв. № дубл.	Подп. и дата	3271.102092.000 ПЗ					Лист
											50
Изм.	Лист	№ докум.	Подпись	Дата							

Таблица 8 – Сравнительный анализ *Java*, *JavaScript* и *Python*

	<i>Java</i>	<i>JavaScript</i>	<i>Python</i>
Предрасположенность к веб-разработке	Отсутствует	Присутствует	Отсутствует
Наличие среды разработки	Необходимо	Необязательно	Необязательно
Проверка кода компилятором	Присутствует	Отсутствует	Отсутствует
Направление использования	Бэкенд (мобильные приложения, компьютерные программы, игры и т.д.)	Программы, работающие в браузере (анимации, алгоритмы), мобильные приложения	Бэкенд (мобильные приложения, компьютерные программы, игры и т.д.)
Наличие необходимых для проекта фреймворков, библиотек ориентированных на веб-разработку	Отсутствует	Присутствуют	Отсутствуют
Поддержка крупных компаний для дальнейшей разработки в веб-сегменте	Отсутствует	Присутствуют	Отсутствуют

В программе, написанной на языке *Java*, компилятор проводит синтаксический и семантический анализ, и благодаря этому имеется возможность исправления ошибок до запуска приложения. Но в случае отсутствия ошибок в коде программы, данная особенность языка замедляет работу всего приложения при каждом удачном запуске [12].

Отсутствие компилятора в языке *JavaScript* предполагает отсутствие возможности проверки кода до запуска программы, что в случаях многократного успешного выполнения программного кода, предоставляет максимально возможную скорость работы приложения, являясь первостепенным значением в генерации штрихкодов, в частности при обработке множества входных данных [26].

Инв. № подл.	Подп. и дата
Инв. № дубл.	Взам. инв. №
Подп. и дата	Подп. и дата

Изм.	Лист	№ докум.	Подпись	Дата	3271.102092.000 ПЗ	Лист
						51

Для использования, редактирования кода программ на *JavaScript*, *Python* достаточно возможностей встроенного в операционную систему текстового редактора, что является определенным преимуществом, так как для работы с языком *Java* необходима среда разработки, как правило требующая затрат на лицензию и временных ресурсов для обучения по использованию среды разработки и ее установке на машине [32].

Разрабатываемый код проекта работает в условиях возможностей браузеров пользователя, а так как возможности языка *JavaScript* больше ориентированы на разработку программ для браузеров, нежели *Java* или *Python*, то будет использоваться данный язык программирования [25].

Техническая часть проекта реализована при помощи инструмента разработки под названием *ReactJS*. *ReactJS* – это библиотека для языка программирования *JavaScript* ориентированная на разработку пользовательских интерфейсов. Сравнительная часть представлена в таблице 9.

Таблица 9 – Сравнительная таблица *ReactJS*, *AngularJS*

	<i>ReactJS</i>	<i>AngularJS</i>
Привязка данных	Односторонняя привязка данных означает, что элемент пользовательского интерфейса не может влиять на состояние компонента.	Поддерживает как одностороннюю, так и двустороннюю привязку данных, двусторонняя привязка данных означает, что если мы изменим входные данные пользовательского интерфейса, состояние модели изменится, и наоборот.
Базовый язык для разработки	<i>JavaScript</i>	<i>TypeScript</i>
Компоненты пользовательского интерфейса	Инструменты пользовательского интерфейса, разработанные сообществом, предоставляют широкий спектр компонентов пользовательского интерфейса.	<i>Angular</i> включает в себя ряд компонентов <i>material design</i> , которые упрощают настройку пользовательского интерфейса.

Инв. № подл.	Подп. и дата	Инв. № дубл.	Взам. инв. №	Подп. и дата						Лист
Изм.	Лист	№ докум.	Подпись	Дата	3271.102092.000 ПЗ					52

Продолжение таблицы 9

<i>DOM</i>	Виртуальный <i>DOM</i> – каждый раз, когда <i>DOM</i> изменяется, создается новый виртуальный <i>DOM</i> по сравнению с предыдущим, и в «реальном» <i>DOM</i> изменяются только различия.	Инкрементный <i>DOM</i> – когда создается новый <i>DOM</i> , он сравнивает его с предыдущим и применяет различия к «фактическому» <i>DOM</i> , выделяя память только при необходимости.
Направленность разработки	Библиотека связана только с компонентами пользовательского интерфейса.	Полнофункциональный фреймворк – дает представление о том, как должно быть спроектировано ваше приложение, а также ряд библиотек, которые помогают в разработке сложных приложений.
Архитектура кода	Не требует определенной структуры проекта, и вы можете начать использовать его всего с нескольких строк кода. Каждый проект является приложением, файлы которого выполняют свою функцию	Проекты в <i>Angular</i> структурированы на Модули, Компоненты и Сервисы. Каждое приложение <i>Angular</i> имеет как минимум один рутовый компонент и один рутовый модуль.

Библиотека *ReactJS* предоставляет разработчику широкий спектр возможностей в создании динамических страниц, которые облегчают и ускоряют разработку на языке *JavaScript* [20].

Объектная модель документа (*DOM*) – программный интерфейс, предоставляющий программам доступ к *HTML*-документам, а также возможности для изменения и оформления содержимого документов. В используемой библиотеке задействована технология виртуальный *DOM*. Данная технология предоставляет высокую эффективность и производительность, что является весомым преимуществом в сравнении с конкурентами данной библиотеки. Эффективность заключается в использовании двух версий виртуальной *DOM*. Одна из версий является копией настоящего *HTML*-документа, а вторая является копией с присутствием изменений в структуре документа – обновленной. В дальнейшем

Инв. № подл.	Подп. и дата	Взам. инв. №	Подп. и дата
Инв. № дубл.			

Изм.	Лист	№ докум.	Подпись	Дата	3271.102092.000 ПЗ	Лист
						53

сравниваются данные версии документа между собой для поиска измененных элементов в структуре документа, и своевременное изменение части реального *DOM*, а именно настоящего *HTML*-файла. Данная технология является эффективной, так как нет необходимости в обновлении целого *HTML*-документа, а только его части, что экономит ресурсы системы в ходе загрузки, и, следовательно, влияет на улучшение производительности всей системы [21].

Огромным преимуществом также является архитектура создаваемого приложения на базе *ReactJS*. Каждый компонент в приложении выполняет свою определенную функцию, которую разработчик реализует при помощи программного кода.

По сравнению с фреймворками-конкурентами, например, *AngularJS*, данная особенность предоставляет возможности для ускоренного изучения, разработки, отладки и тестирования продукта, благодаря разделению большого приложения на мелкие части.

Основная часть разработки проекта относится к реализации пользовательского интерфейса. Превалирующая часть компонентов библиотеки *ReactJS* предрасположена к разработке пользовательского интерфейса, а именно реализация последнего является основной частью работы в создании дипломного проекта.

В отличие от вышеуказанного фреймворка *Angular*, основным языком разработки которого является *TypeScript*, одним из достоинств в пользу выбора *React*, является его основной язык разработки – *JavaScript*, так как нет необходимости в изучении дополнительных языков программирования. Благодаря доступности документации для изучения библиотеки *React*, имеется возможность получения необходимого опыта и знаний для создания проекта в кратчайшие сроки [22].

В данном проекте используется реляционная система управления базами данных *MySQL*. Данная система широко популярная в среде разработки веб-приложений благодаря своей простоте, надежности и ориентированности на

Инв. № подл.	Подп. и дата	Инв. № дубл.	Взам. инв. №	Подп. и дата	Инв. № подл.					Лист
Изм.	Лист	№ докум.	Подпись	Дата	3271.102092.000 ПЗ				54	



возможности разработчиков. Для выявления особенностей данного СУБД будет использован сравнительный анализ возможностей *MySQL* с функциональными возможностями *PostgreSQL*. Сравнительный анализ *MySQL* представлен в таблице 10.

Таблица 10 – Сравнительный анализ *PostgreSQL* и *MySQL*

	<i>PostgreSQL</i>	<i>MySQL</i>
Направление использования	Большие и сложные аналитические процессы [33].	Банкинг, сложные многостраничные веб-сайты
Репликация	При физической репликации на сервер реплики передается поток записей. Логическая репликация работает по принципу модели клиент-сервер.	Масштабируемость и производительность приложений с помощью различных типов синхронизации.
Масштабирование	Известна своей вертикальной масштабируемостью. Можно обрабатывать крупные объемы данных и манипулировать вычислительными мощностями.	Используется в случае горизонтальной масштабируемости. Возможность масштабировать увеличением новых узлов в кластере базы данных. Подходит для веб-приложений с настройками на обработку небольшого количества данных.
Производительность	Создана как универсальная система, способная справляться как с нагрузками при интенсивных операциях чтения, так и при интенсивной записи. Но за счет этого выдается меньшая производительность.	Оптимизирована для нагрузок с непрерывными операциями чтения данных, что помогает улучшить производительность запросов.
Хранимые процедуры	Поддерживает хранимые процедуры, написанные на различных языках, включая <i>PL/pgSQL</i> , <i>PL/Tcl</i> , <i>PL/Perl</i> .	<i>MySQL</i> поддерживает хранимые процедуры, написанные на языке <i>SQL</i> .

Инв. № подл.	Подп. и дата
Инв. № дубл.	Взам. инв. №
Подп. и дата	
Инв. № подл.	

*MySQL* – это система управления базами данных в веб-среде. Функции данного СУБД необходимо реализовывать в разработке веб-приложений, так как данная система эффективно проявляет себя с небольшими объемами данных [13].

*PostgreSQL* является широкопрофильной системой управления базами данных. Благодаря богатому функционалу программы, каждый пользователь имеет возможность в реализации своего проекта любого уровня сложности [19].

Несмотря на преимущества данного СУБД, не каждый пользователь имеет возможность изучить предоставляемый инструментарий приложения и сталкивается с затруднениями при изучении необходимых пользователю функций.

*MySQL* использует ограниченный набор типов данных, благодаря чему является более ориентированным для использования в небольших программах. В отличие от него, *PostgreSQL* будучи ориентированным на использование в крупных системах, перегружен поддержкой современных типов данных, что не является необходимостью в использовании небольших приложений, а скорее ограничением.

Часто загруженность данной системы уменьшает скорость обработки данных в ежедневных сценариях работы, что является недостатком для большого числа пользователей, и недопустимым характером выполнения задачи в дипломном проекте. В генерации штрихкодов важным качеством является скорость выполнения алгоритма преобразования полученных данных в графический вид, поэтому имеется необходимость в использовании данных без задержек. Система управления базами данных *MySQL* предоставляет именно такие возможности [15].

Все перечисленное программное обеспечение необходимо для успешного выполнения поставленных целей и задач, так как каждое из них предоставляет определенный спектр возможностей

Инв. № подл.	Подп. и дата	Инв. № дубл.	Взам. инв. №	Подп. и дата						Лист
										56
					Изм.	Лист	№ докум.	Подпись	Дата	3271.102092.000 ПЗ

### 3.2 Руководство пользователя

Одна из ключевых задач пользовательского интерфейса системы – упрощение взаимодействия пользователя со сложными техническими объектами. Это означает, что пользователь должен иметь право пользования всеми техническими особенностями системы и ее функциями, без каких-либо затруднений [18].

В ходе проектирования модуля *ERP*-системы, был определен ряд требований к пользовательскому интерфейсу:

1) Эргономичность. Нацеленность на создание условий, в которых задача выполняется при совершении наименьшего количества действий от пользователя. Например, для печати документа достаточно нажать кнопку «Печать».

2) Проработанные сценарии ошибок. Для избегания пользователя встречи с неопределенными ситуациями в процессе работы, необходимо выявить все неисправности в работе системы и исправить их.

3) Эффективность. Программное обеспечение должно выполнять функции, соответствующие требованиям пользователя [34].

В ходе разработки был создан интерфейс модуля генератора штрихкодов *ERP*-системы в соответствии с рисунком 10:

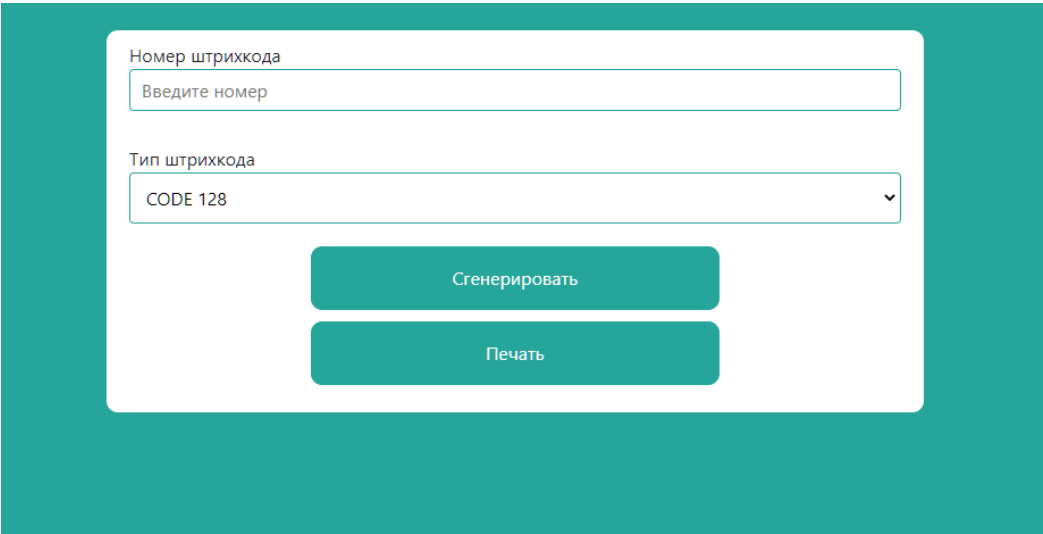


Рисунок 10 – Стартовый вид страницы

Подп. и дата	
Взам. инв. №	
Инв. № дубл.	
Подп. и дата	
Инв. № подл.	

Изм.	Лист	№ докум.	Подпись	Дата

Для корректного пользования программой, необходимо выполнять данный порядок действий:

1) Ввод входных данных.

Принцип ввода изображен на рисунке 11.

A screenshot of a web application interface for generating barcodes. It features a teal background. A white form contains two input fields: 'Номер штрихкода' (Barcode Number) with the value '23232323' and 'Тип штрихкода' (Barcode Type) with a dropdown menu showing 'CODE 128'. Below the fields are two teal buttons: 'Сгенерировать' (Generate) and 'Печать' (Print).

Рисунок 11 – Ввод входных данных

2) Выбор типа штрихкода из выпадающего списка в соответствии с рисунком 12. В данный список входят такие типы: *code-128, codabar, code-39, ean-2, ean-5, ean-8, ean-13, upc, itf-14, msi, pharmacode, data matrix, qr-code*.

A screenshot of the same web application interface as Figure 11, but with the 'Тип штрихкода' dropdown menu open. The menu lists various barcode types: CODE 128, Codabar, CODE 39, EAN 2, EAN 5, EAN 8, EAN 13, UPC, ITF-14, MSI, Pharmacode, Data Matrix, and QR-Code. 'Codabar' is highlighted in blue.

Рисунок 12 – Выбор типа штрихкода

Подп. и дата	
Взам. инв. №	
Инв. № дубл.	
Подп. и дата	
Инв. № подл.	

Изм.	Лист	№ докум.	Подпись	Дата

3) Нажать на кнопку: «Сгенерировать» для генерации штрихкода по введенным данным в соответствии с рисунком 13.



Рисунок 13 – Генерация по кнопке

4) Отправка штрихкода на печать. Для этого необходимо нажать на кнопку «Печать» в соответствии с рисунком 14.

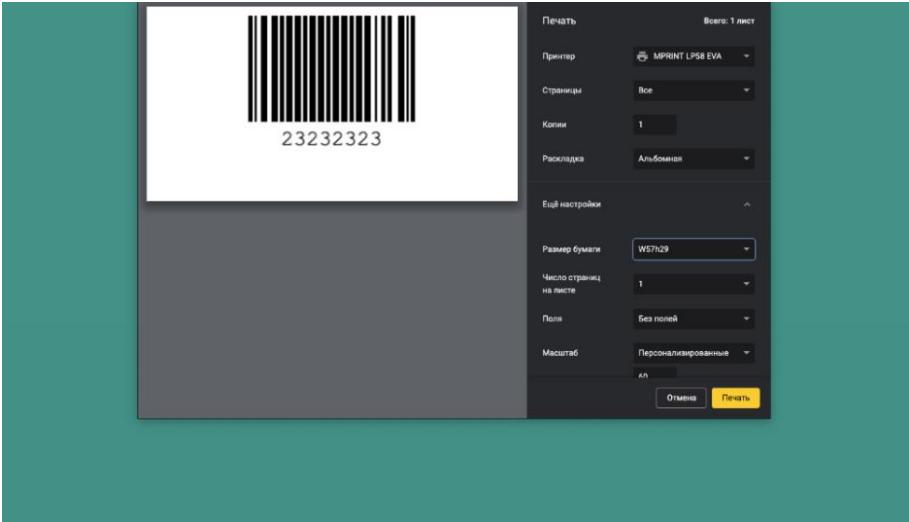


Рисунок 14 – Открытие окна печати штрихкода

Данный интерфейс выполняет все необходимые задачи для полноценной реализации системы.

Инв. № подл.	Подп. и дата	Взам. инв. №	Подп. и дата

### 3.3 Руководство программиста

#### 3.3.1 Режим работы программы

Web-интерфейс в браузере (с поддержкой *HTML5*)

#### 3.3.2 Средства проверки правильности выполнения программы

Корректность и исправность программного обеспечения проверяется в ходе выполнения подготовленных заданий и примеров.

При неправильном вводе данных выходят соответствующие ошибки, изображенной на рисунке 15.

Одной из таких является неправильное количество возможных символов для данного типа штрихкода, так как для некоторых типов штрихкодов важно использование определенного количества символов.

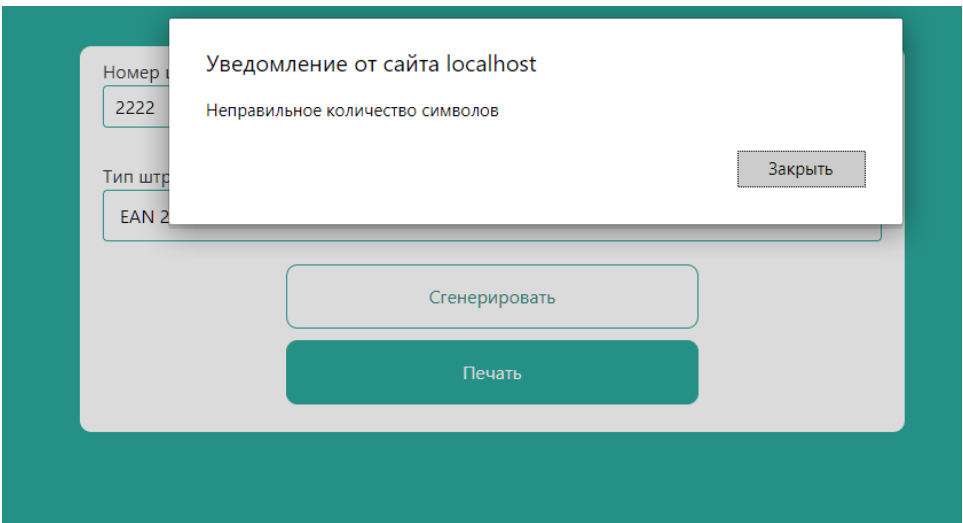


Рисунок 15 – Проверка ошибки неправильного количества символов

Также некорректное использование символов, например, использование латиницы в штрихкодах, где возможно использовать только цифры, провоцирует ошибку в соответствии с рисунком 16.

Подп. и дата	
Взам. инв. №	
Инв. № дубл.	
Подп. и дата	
Инв. № подл.	

					3271.102092.000 ПЗ
Изм.	Лист	№ докум.	Подпись	Дата	

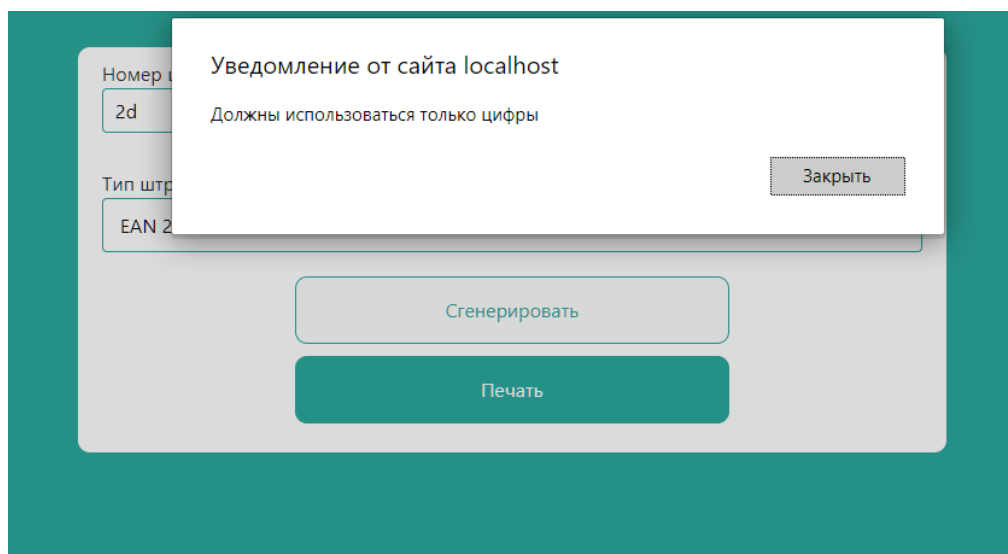


Рисунок 16 – Ошибка использования некорректных символов

При попытке аварийного прекращения работы программы, браузер выполняет обновление страницы для повторной загрузки программного кода и устранения причины сбоя. При этом введенные или полученные данные будут утеряны. В случае же повторения аварийной ситуации, предоставляется информация об ошибке.

### 3.3.3 Входные и выходные данные

В качестве входной информации при использовании модуля генерации штрихкодов должны использоваться:

- цифровые данные – информация, представленная в виде числового формата;
- цифро-буквенные значения – информация, представленная в виде сочетания числового и буквенного формата данных;
- буквенный формат – информация, представленная в виде буквенного формата. Программа заточена на использование латиницы.

Также необходимо выбрать тип штрихкода из представленных в выпадающем списке. По умолчанию используется: «code-128». Для каждого из

Инв. № подл.	Подп. и дата
Инв. № дубл.	Взам. инв. №
Подп. и дата	Подп. и дата
Инв. № подл.	Подп. и дата

Изм.	Лист	№ докум.	Подпись	Дата	3271.102092.000 ПЗ	Лист
						61

типов штрихкодов существуют ограничения, необходимые для корректной работы программного обеспечения.

Данным ограничением является количество символов, возможное для использования в каждом конкретном формате штрихкода, например:

- *CODE-128*: до 128 символов;
- *CODABAR*: до 128 символов;
- *CODE-39*: до 128 символов;
- *EAN-2*: 2 символа;
- *EAN-5*: 5 символов;
- *EAN-8*: 8 символов;
- *EAN-13*: 12 символов;
- *UPC*: 11 символов;
- *ITF-14*: 13 символов;
- *MSI*: до 128 символов;
- *Pharmacode*: 5 символов;
- *Data Matrix*: до 128 символов;
- *QR-Code*: до 128 символов.

В качестве выходной информации выступает:

- сгенерированный штрихкод;
- передача на печать.

Выходная информация предоставляется в графическом виде, как в бумажном, так и электронном вариантах.

### 3.3.4 Описание основных функций

1) Генерация штрихкодов выбранного типа на основе входных данных происходит в соответствии с примером на рисунке 17. Основной задачей данного программного обеспечения является генерация штрихкодов по выбранному типу. Данный модуль построен в концепции *React JavaScript*.

Инв. № подл.	Подп. и дата	Инв. № дубл.	Взам. инв. №	Подп. и дата						Лист
Изм.	Лист	№ докум.	Подпись	Дата	3271.102092.000 ПЗ					62



Номер штрихкода

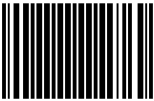
23232323

Тип штрихкода

CODE 128

Сгенерировать

Печать



23232323

Рисунок 17 – Пример генерации

Данный алгоритм можно разделить на несколько этапов:

- 1) получение данных;
- 2) выбор типа штрихкода;
- 3) генерация штрихкодов.

```
function Barcode({articuls,types}) {
  const { inputRef } = useBarcode({
    value: articuls,
    options:{
      format: types,
    }
  });
  return <img ref={inputRef} alt=""/>;
};
```

```
function Barcodes() {
  let articul = React.createRef();
  let type = React.createRef();
  const [articuls, setArticuls] = useState();
  const [types, setTypes] = useState();
```

```
const setInputToValue = () => {
  let artCurrent = articul.current.value;
  let typeCurrent = type.current.value;
  if (artCurrent.length !== symbolsKol[typeCurrent])
  {
    if (artCurrent.length > symbolsKol[typeCurrent]){
      return alert('Неправильное количество символов');
```

Инв. № подл.	Подп. и дата	Взам. инв. №	Подп. и дата
Инв. № дубл.			

Изм.	Лист	№ докум.	Подпись	Дата
------	------	----------	---------	------

3271.102092.000 ПЗ

Инв. № подл.	Подп. и дата	Инв. № дубл.	Взам. инв. №	Подп. и дата

```

        ? <QRCode value={articuls}/>
        : (types === 'datamatrix' ? <DataMatrix msg={articuls} pal="" /> : <Barcode
articuls={articuls} types={types}/>)
    }
  </div>
</div>
</div>
);}

```

2) Передача на печать сгенерированных штрихкодов. Одной из основных задач данного программного обеспечения является печать сгенерированных штрихкодов. Данная функция необходима для полноценной реализации процесс получения штрихкода.

Данный алгоритм можно разделить на несколько этапов:

- 1) получение штрихкода;
- 2) открытие окна печати браузера;
- 3) заготовка шаблона для печати;
- 4) передача на печать в принтер/термопринтер.

```

const componentRef= useRef();
const generatePDF = useReactToPrint({
  content: ()=>componentRef.current,
  documentTitle: "Печать штрихкода",
});
<div id='print-barcode' ref={componentRef} style={{width:'100%', display:'flex',
justifyContent:'center'}}>
  <div>
    {
      types === 'QRCode'
      ? <QRCode value={articuls}/>
      : (types === 'datamatrix' ? <DataMatrix msg={articuls} pal="" /> : <Barcode
articuls={articuls} types={types}/>)
    }
  </div>
</div>

```

В данном программном обеспечении отсутствует функционал вывода сообщения специально для программиста. Для получения необходимой информации в ходе работы программы появляются общие сообщения, сгенерированные браузером или операционной системой.

Подп. и дата	
Взам. инв. №	
Инв. № дубл.	
Подп. и дата	
Инв. № подл.	

					3271.102092.000 ПЗ	Лист
Изм.	Лист	№ докум.	Подпись	Дата		65

## ЗАКЛЮЧЕНИЕ

В ходе выполнения выпускной квалификационной работы был разработан и реализован модуль генерации штрихкодов для *ERP*-системы, необходимый для повышения эффективности складского учета. Изучены некоторые области использования и их особенности. Подробно изучен процесс приемки и штрихкодирования товара из области учета склада, на примере которого был изучен алгоритм работы программного модуля.

Целью данной выпускной квалификационной работы являлось повышение эффективности складского учета при помощи модуля генерации штрихкодов для *ERP*-системы. Поставленная цель была достигнута при помощи поэтапного выполнения задач. Во время проведения анализа предметной области были выявлены области использования технологии штрихкодирования, одной из которых является складской учет. Был изучен и смоделирован бизнес-процесс штрихкодирования товара и передачи его покупателю в виде мнемосхемы, а также, представлена его часть в нотации *BPMN* – процесс приемки и штрихкодирования товара. Были спроектированы и разработаны логические и физические модели базы данных. Для полноценного представления расположения модуля в *ERP*-системе спроектирована и смоделирована архитектура системы вместе с разрабатываемым модулем. Также был разработан модуль генерации штрихкодов для *ERP*-системы с дополнительной функцией печати, а также разработаны руководства программиста и пользователя, которые предоставляют информацию по использованию модуля.

Инв. № подл.	Подп. и дата	Взам. инв. №	Подп. и дата	Инв. № дубл.	Подп. и дата	

					3271.102092.000 ПЗ	Лист
Изм.	Лист	№ докум.	Подпись	Дата		66

## СПИСОК ЛИТЕРАТУРЫ

1. Штриховое кодирование: учебное пособие / А. А. Востриков, А. М. Сергеев - Санкт-Петербург: ГУАП, 2010. — 56 с.
2. Грошев, А. С. Информатика: учебник / А. С. Грошев, П. В. Закляков. — 3-е изд. — Москва: ДМК Пресс, 2015. — 588 с.
3. Щербакова, Е. В. Инновационные технологии в хранении / Е. В. Щербакова, Е. А. Ольховатов, А. В. Степовой. — Санкт-Петербург: Лань, 2023. — 168 с.
4. Документооборот и делопроизводство: методические указания / составители О. А. Красненкова, М. К. Рожко. — Санкт-Петербург: СПбГУ ГА, 2022. — 34 с.
5. Одинцова, М. А. Информационные системы управления ресурсами предприятия: учебное пособие / М. А. Одинцова. — Москва: РТУ МИРЭА, 2022. — 187 с.
6. Черных, В. В. *ERP*-системы управления производственным предприятием: практикум : учебное пособие / В. В. Черных. — Йошкар-Ола: ПГТУ, 2018. — 64 с.
7. Зуева, А. Н. Моделирование бизнес-процессов в нотации *BPMLN 2.0*: учебное пособие / А. Н. Зуева. — Москва: РТУ МИРЭА, 2021. — 105 с.
8. Королев, А. С. Проектирование логической архитектуры систем с использованием средств модели-ориентированной системной инженерии: учебно-методическое пособие / А. С. Королев. — Москва: РТУ МИРЭА, 2020. — 40 с.
9. Федотов, А. В. Компьютерное управление в производственных системах: учебное пособие для вузов / А. В. Федотов, В. Г. Хомченко. — 2-е изд., стер. — Санкт-Петербург: Лань, 2021. — 620 с.
10. Эделман, Д. Автоматизация программируемых сетей: руководство / Д. Эделман, С. С. Лоу, М. Осуолт; перевод с английского А. В. Снастина. — Москва: ДМК Пресс, 2019. — 616 с.
11. Бурков, А. Инженерия машинного обучения / А. Бурков ; перевод с английского А. А. Слинкина. — Москва : ДМК Пресс, 2022. — 306 с.

Инв. № подл.	Подп. и дата	Взам. инв. №	Подп. и дата	7. Зуева, А. Н. Моделирование бизнес-процессов в нотации BPMN 2.0: учебное пособие / А. Н. Зуева. — Москва: РТУ МИРЭА, 2021. — 105 с.
				8. Королев, А. С. Проектирование логической архитектуры систем с использованием средств модели-ориентированной системной инженерии: учебно-методическое пособие / А. С. Королев. — Москва: РТУ МИРЭА, 2020. — 40 с.
Инв. № подл.	Подп. и дата	Взам. инв. №	Подп. и дата	9. Федотов, А. В. Компьютерное управление в производственных системах: учебное пособие для вузов / А. В. Федотов, В. Г. Хомченко. — 2-е изд., стер. — Санкт-Петербург: Лань, 2021. — 620 с.
				10. Эделман, Д. Автоматизация программируемых сетей: руководство / Д. Эделман, С. С. Лоу, М. Осуолт; перевод с английского А. В. Снастина. — Москва: ДМК Пресс, 2019. — 616 с.
Инв. № подл.	Подп. и дата	Взам. инв. №	Подп. и дата	11. Бурков, А. Инженерия машинного обучения / А. Бурков ; перевод с английского А. А. Слинкина. — Москва : ДМК Пресс, 2022. — 306 с.
Изм.	Лист	№ докум.	Подпись	Дата

3271.102092.000 ПЗ

Лист 67

12. Фримен, Э. Изучаем *HTML*, *XHTML* и *CSS (Head First)* / Э. Фримен, Э. Фримен — СПб.: Питер, 2012. — 720 с.
13. Хромушин, В. А. Сборник примеров *HTML* страниц: учебное пособие / В. А. Хромушин, Р. В. Грачев, Н. Д. Юдакова. — Тула: ТулГУ, 2022. — 192 с.
14. Дроговоз, П. А. Управление проектами в системе *SAP ERP* : учебное пособие / П. А. Дроговоз, Л. И. Садовский, П. Д. Иванов. — Москва : МГТУ им. Н.Э. Баумана, 2015. — 76 с.
15. Шварц, Б. *MySQL* по максимуму / Б. Шварц, П. Зайцев — СПб.: Питер, 2018. — 864 с.
16. Кириллина, Ю. В. Управление бизнес-процессами: учебное пособие / Ю. В. Кириллина. — Москва: РТУ МИРЭА, 2022. — 159 с.
17. Кириллина, Ю. В. Реинжиниринг бизнес-процессов: методические рекомендации / Ю. В. Кириллина. — Москва: РТУ МИРЭА, 2021. — 31 с.
18. Малышев, К. В. Построение пользовательских интерфейсов / К. В. Малышев. — Москва: ДМК Пресс, 2021. — 268 с.
19. Осипов, Д. Л. Технологии проектирования баз данных / Д. Л. Осипов. — Москва: ДМК Пресс, 2019. — 498 с.
20. Хантер, I. Т. Многопоточный *JavaScript* / I. Т. Хантер, Б. Инглиш; перевод с английского А. А. Слинкина. — Москва: ДМК Пресс, 2022. — 188 с.
21. Диков, А. В. Клиентские технологии веб-программирования: *JavaScript* и *DOM*: учебное пособие / А. В. Диков. — Санкт-Петербург: Лань, 2020. — 124 с.
22. Дилеман, П. Изучаем *Angular 2* / П. Дилеман; под редакцией А. Н. Киселева; перевод с английского Р. Н. Рагимова. — Москва: ДМК Пресс, 2017. — 354 с.
23. Лоре, А. Проектирование веб-API: руководство / А. Лоре; перевод с английского Д. А. Беликова. — Москва: ДМК Пресс, 2020. — 440 с.
24. Складской учет [Электронный ресурс] Источник: <https://www.klerk.ru/blogs/moysklad/529242/>
25. *Java* и *JavaScript* [Электронный ресурс] Источник: <https://practicum.yandex.ru/blog/chem-otlichayutsya-java-i-javascript/>

Инв. № подл.	Подп. и дата	Инв. № дубл.	Взам. инв. №	Подп. и дата	<div>3271.102092.000 ПЗ</div>	Лист
						68
						Изм. Лист № докум. Подпись Дата



## ПРИЛОЖЕНИЕ А

### Листинг кода

```
import React, { useState, useRef } from 'react';

import { useBarcode } from 'next-barcode';

import QRCode from 'qrcode.react'

import DataMatrix from 'react-datamatrix-svg';

import { useReactToPrint } from 'react-to-print';
```

```
function Barcode({ articuls,types }) {

  const { inputRef } = useBarcode({

    value: articuls,

    options:{

      format: types,

    }

  })

  return <img ref={inputRef} alt=""/>;

};
```

```
function Barcodes() {

  let articul = React.createRef();

  let type = React.createRef();

  const [articuls, setArticuls] = useState();

  const [types, setTypes] = useState();
```

Инв. № подл.	Подп. и дата	Взам. инв. №	Подп. и дата
Инв. № дубл.			
Подп. и дата			

					3271.102092.000 ПЗ	Лист
						70
Изм.	Лист	№ докум.	Подпись	Дата		



```

var symbolsKol = {

    'code128':128,

    'codabar':128,

    'code39': 128,

    'ean2': 2,

    'ean5': 5,

    'ean8': 7,

    'ean13': 12,

    'upc': 11,

    'itf14': 13,

    'msi':128,

    'pharmacode':5,

    'datamatrix':128,

    'QRCode':128

};

function isNumber(e){

    return /^d+$/.test(e);

}

const setInputToValue = () => {

    let artCurrent = articul.current.value;

    let typeCurrent = type.current.value;

    if (artCurrent.length !== symbolsKol[typeCurrent])

```

Инв. № подл.	Подп. и дата	Взам. инв. №	Подп. и дата
Инв. № дубл.			
Подп. и дата			
Инв. № подл.			

Изм.	Лист	№ докум.	Подпись	Дата	3271.102092.000 ПЗ	Лист
						71

```

{
  if (artCurrent.length > symbolsKol[typeCurrent]){
    return alert('Неправильное количество символов');
  }

  else if (symbolsKol[typeCurrent] < 128) {
    return alert('Неправильное количество символов');
  }
}

if ((isNumber(artCurrent) === false) && ((typeCurrent === 'ean5' ||
(typeCurrent === 'ean2') || (typeCurrent === 'ean8')

|| (typeCurrent === 'codabar') || (typeCurrent === 'ean13') || (typeCurrent ===
'upc') || (typeCurrent === 'itf14')

|| (typeCurrent === 'itf') ||(typeCurrent === 'msi')))) {
  return alert ('Должны использоваться только цифры');
}

setArticuls(artCurrent);

setTypes(typeCurrent);

};

const componentRef= useRef();

const generatePDF = useReactToPrint({
  content: ()=>componentRef.current,
  documentTitle: "Печать штрихкода",
});

```

Инв. № подл.	Подп. и дата
Инв. № дубл.	Взам. инв. №
Подп. и дата	
Инв. № подл.	

Изм.	Лист	№ докум.	Подпись	Дата	3271.102092.000 ПЗ	Лист
						72

return (

<div className="container">

<div className="form\_articul">

<label className='articul\_label'>Номер штрихкода</label>

<input className='articul\_input' placeholder="Введите номер"  
ref={ articul } defaultValue=""/>

</div>

<div className="form\_type">

<label className="banner text-center">Тип штрихкода</label>

<select id="barcodeType" className="form\_control" ref={ type }>

<option value="code128">CODE 128</option>

<option value="codabar">Codabar</option>

<option value="code39">CODE 39</option>

<option value="ean2">EAN 2</option>

<option value="ean5">EAN 5</option>

<option value="ean8">EAN 8</option>

<option value="ean13">EAN 13</option>

<option value="upc">UPC</option>

<option value="itf14">ITF-14</option>

<option value="msi">MSI</option>

<option value="pharmacode">Pharmacode</option>

<option value="datamatrix">Data Matrix</option>

Подп. и дата	
Взам. инв. №	
Инв. № дубл.	
Подп. и дата	
Инв. № подл.	

Изм.	Лист	№ докум.	Подпись	Дата

3271.102092.000 ПЗ

<option value="QRCode">QR-Code</option>

</select>

</div>

<button className="btn generate"  
onClick={setInputToValue}>Сгенерировать</button>

<button className="btn print" onClick={generatePDF}>Печать</button>

<div id='print-barcode' ref={componentRef} style={{ width:'100%',  
display:'flex', justifyContent:'center'}}>

<div>

{

types === 'QRCode'

? <QRCode value={articuls}/>

: (types === 'datamatrix' ? <DataMatrix msg={articuls} pal=""/> : <Barcode  
articuls={articuls} types={types}/>)

}

</div>

</div>

</div>

);

}

export default Barcodes;

Подп. и дата	
Взам. инв. №	
Инв. № дубл.	
Подп. и дата	
Инв. № подл.	

					3271.102092.000 ПЗ	Лист
						74
Изм.	Лист	№ докум.	Подпись	Дата		