Министерство образования и науки Российской Федерации
Федеральное государственное бюджетное образовательное учреждение
высшего профессионального образования
«Южно-Уральский государственный университет»
(национальный исследовательский университет)
Факультет «Компьютерных технологий, управления и радиоэлектроники»
Кафедра «Электронные вычислительные машины»

ДОПУСТИТЬ К ЗАЩИТЕ
Заведующий кафедрой
_____К. А. Домбровский
«___»_____2016г.

Система контроля объектов в ограниченном объеме

ПОЯСНИТЕЛЬНАЯ ЗАПИСКА
К ВЫПУСКНОЙ КВАЛИФИКАЦИОННОЙ РАБОТЕ
ЮУрГУ – 230100.2016.227 ПЗ ВКР

Руководитель работы,
доцент каф. «Электронные
Вычислительные машины»
_____ И. Л. Кафтанников
«___»_____2016г.

Автор работы
Студенты группы КТУР-417
_____ Г. Р. Зарипова
«05» июня 2016г.

_____ Е. А. Кучеров
«05» июня 2016г.

Нормоконтролер, ст. преп. каф.
«Электронные Вычислительные
машины»
_____ В. В. Лурье
«05» июня 2016г

Челябинск 2016

Министерство образования и науки Российской Федерации
Федеральное государственное бюджетное образовательное учреждение
высшего профессионального образования
«Южно-Уральский государственный университет»
(национальный исследовательский университет)
Факультет «Компьютерных технологий, управления и радиоэлектроники»
Кафедра «Электронные вычислительные машины»

УТВЕРЖДАЮ
Заведующий кафедрой
_____ К. А. Домбровский
«___»_____2016г.

З А Д А Н И Е

на выпускную квалификационную работу студентов

Зариповой Гульназ Раисовны, Кучерова Евгения Андреевича

Группа КТУР – 417

1 Тема работы
Разработка системы контроля объектов в ограниченном объеме
утверждена приказом по университету от _____2016 г. № _____
2 Срок сдачи студентом законченной работы _____2016 г.
3 Исходные данные к работе
Вычислительная концепция «Интернет вещей».

4 Содержание расчетно-пояснительной записки

Введение
1 Анализ системы
2 Проектирование системы
3 Разработка системы
4 Руководство пользователя системы
5 Введение системы в эксплуатацию
6 Технико-экономические показатели
Заключение

5 Перечень графического материала

Концепция IOT. Цели создания системы. Требования к системе – 1;
Обзор и сравнительный анализ технологий – 3;
Система контроля объектов для жилых помещений – 3;
Система контроля объектов для складских помещений – 3.

Всего 9 листов

6 Консультанты по работе, с указанием относящихся к ним разделов работы

Раздел	Консультант	Подпись, дата	
		Задание выдал	Задание принял
-	-	-	-

7 Дата выдачи задания 16.01.2016

Руководитель _____ Ксенофонтов С.Л.
(подпись) (И.О.Ф.)

Задание принял к исполнению _____ / Г.Р. Заринова
(подпись студента) (И.О.Ф.)

_____ / Е.А. Кучеров

КАЛЕНДАРНЫЙ ПЛАН

Наименование этапов выпускной квалификационной работы	Срок выполнения этапов работы	Отметка о выполне руководителя
Анализ задачи	14.02.2016	
Обзор технологий	5.04.2016	
Разработка БД	10.04.2016	
Написание ПО	15.04.2016	
Оформление работы	26.05.2016	

Заведующий кафедрой _____ / И.О.Ф Блинова

Руководитель работы _____ / И.О.Ф Карбаинов

Студент _____ / И.О.Ф Т.Р. Зарипо

Студент _____ / И.О.Ф Е.А. Кучер

АННОТАЦИЯ

Зарипова Г.Р. Кучеров Е.А. Разработка системы контроля объектов в ограниченном объеме. – Челябинск: ФГБОУ ВПО «ЮУРГУ» (НИУ), КТУР; 2016, 115 с., 48 ил., библиогр. список – 25 наим., 2 прил., 30 слайдов.

После анализа вычислительной концепции «Интернет вещей», была предложена разработка системы контроля объектов на основе радиочастотной идентификации.

Была разработана база данных и написан интерфейс приложения системы идентификации объектов для взаимодействия компонентов системы.

Внедрение данной системы оптимизирует рабочее время производственного процесса за счет отслеживания перемещений объектов, а также сокращает время пользователя на поиск необходимого объекта в жилом помещении.

Изм.	Лист	№ докум.	Подпись	Дат	230100.2016.227/228.00 ПЗ		
Разраб.		Г.Р. Зарипова		05.06	Лит.	Лист	Листов
Разраб.		Е.А. Кучеров		05.06.			
Провер.		ИЛКафтанников		06.06	ФГБОУВПО «ЮУрГУ» (НИУ) Кафедра ЭВМ		
Н. контр.		В. В. Лурье		05.06.16			
Утверд.		КА.Домбровский					

Оглавление

Введение 6

1 Анализ системы 8

1.1 Цели создания системы 8

1.2 Задачи системы 8

1.3 Характеристика объектов автоматизации 8

1.3.1 Характеристика объекта автоматизации системы, используемой единичным пользователем 9

1.3.2 Характеристика объекта автоматизации системы, используемой в промышленной сфере 9

1.4 Актуальность использования системы в жилих помещениях 10

1.5 Определение требований к системе 14

1.5.1 Основные требования к системе, используемой в жилых помещениях 14

1.5.2 Основные требования к системе, используемой в складских помещениях 15

1.6 Анализ структуры компонентов системы 15

1.7 Обзор и сравнительный анализ технологий 16

1.7.1 Штриховое кодирование 16

1.7.2 Технология GPS/ГЛОНАСС 18

1.7.3 Bluetooth-маячки 20

1.7.4 RFID-технология 21

Выводы по разделу один 27

2 Проектирование системы 28

2.1 Основные этапы разработки системы 28

2.2 Описание компонентов системы и выбор проектных решений разработки системы 29
2.2.1 Описание компонентов системы для жилых помещений 29
2.2.1.1 Сервер 30
2.2.1.2 Клиентская часть 30
2.2.2 Описание компонентов системы для складских помещений 31
2.2.2.1 Сервер 31
2.2.2.2 Клиентская часть 31
2.3 Выбор аппаратной части системы 32
2.3.1 Аппаратная часть системы,используемой в жилых помещениях 32
2.3.1.1 RFID-метки 32
2.3.1.2 Считыватель 35
2.3.1.3 Рабочая частота системы 37
2.3.2 Аппаратная часть системы,используемой в складских помещениях 40
2.3.2.1 RFID-метки 40
2.3.2.2 Считыватель 42
2.3.2.3 Принтер RFID-меток 47
2.3.2.4 Рабочая частота системы 48
Выводы по разделу два 49
3 Разработка системы 52
3.1 Создание базы данных 52
3.1.1 Создание базы данных идентификации объектов в жилых помещениях 52
3.1.2 Создание базы данных идентификации объектов в складских помещениях 55

3.2 UML-диаграммы использования системы 59
3.2.1 UML-диаграмма использования системы в жилых помещениях .. 59
3.2.2 UML-диаграмма использования системы в складских помещениях 60
3.3 Разработка интерфейса приложения системы 60
3.3.1 Разработка интерфейса приложения системы, используемой в жилых помещениях 60
3.3.2 Разработка интерфейса приложения системы, используемой в складских помещениях 62
Выводы по разделу три 67
4 Руководство пользователя системы 68
4.1 Руководство пользователя системы, используемой в жилых помещениях 68
4.1.1 Условия эксплуатации 68
4.1.2 Интеграция с существующими системами 68
4.1.3 Обслуживание 69
4.1.4 Эксплуатация 69
4.2 Руководство пользователя системы, используемой в складских помещениях 69
4.2.1 Условия эксплуатации 69
4.2.2 Интеграция с существующими системами 70
4.2.3 Обслуживание 71
4.2.4 Эксплуатация 71
Выводы по разделу четыре 72
5 Введение системы в эксплуатацию 73

5.1 Введение в эксплуатацию системы, используемой в жилых помещениях ……73
5.1.1 Введение в эксплуатацию технических средств 73
5.1.2 Введение в эксплуатацию программного обеспечения..... 73
5.1.3 Обучение пользователей эксплуатации системы..... 73
5.1.4 Опытная эксплуатации..... 73
5.2 Введение в эксплуатацию системы, используемой в складских помещениях 74
5.2.1 Введение в эксплуатацию технических средств 74
5.2.2 Введение в эксплуатацию программного обеспечения..... 74
5.2.3 Обучение пользователей эксплуатации системы 74
5.2.4 Опытная эксплуатация 75
Выводы по разделу пять..... 75
6 Технико-экономические показатели..... 76
6.1 Технико-экономические показатели 76
6.1.1 Примерная калькуляция технических средств системы, используемой в жилых помещениях 76
6.1.2. Примерная калькуляция технических средств системы, используемой в складских помещениях 76
Выводы по разделу шесть 77
Заключение 78
Список литературы 79
ПРИЛОЖЕНИЕ А 81
ПРИЛОЖЕНИЕ Б..... 94

ВВЕДЕНИЕ

В 1926 Никола Тесла в интервью для журнала «Collier’s» сказал, что в будущем радио будет преобразовано в «большой мозг», все вещи станут частью единого целого, а инструменты, благодаря которым это станет возможным, будут легко помещаться в кармане.[3]

Представим, что все окружающие нас предметы (приборы и одежда, автомобили, промышленное оборудование и другое) снабжены идентификационными устройствами. За счет этого генерируется множество данных, которые могут быть использованы различными способами в зависимости от поставленной задачи. Это является основой вычислительной концепции «Интернет вещей», активно развивающейся в наши дни. Интернет вещей (англ. Internet of Things, IoT) — концепция вычислительной сети физических объектов («вещей»), оснащённых встроенными технологиями для взаимодействия друг с другом или с внешней средой. Иными словами, это глобальная сеть компьютеров, датчиков и исполнительных устройств, связывающихся между собой посредством Internet. Ярким примером данной концепции может служить автоматизированная система управления жилым помещением «Умный Дом». Задачей которой является обеспечение комфорта и удобства проживания, путем автоматизации бытовых задач, выполняемых человеком.

Актуальность темы определяется тем, что количество объектов в повседневной жизни человека возрастает и появляется необходимость в их контроле. Немаловажным это становится и в промышленных отраслях, так как учет материальных ценностей становится приоритетной задачей для многих компаний.

Целью дипломной работы является разработка системы контроля объектов в ограниченном объеме, универсальной для единичного пользователя и для использования в промышленной сфере.

Для достижения поставленной цели выделены следующие задачи:

- обзор и сравнительный анализ информационных технологий;
- разработка структуры компонентов системы;
- разработка компонентов программного обеспечения;
- разработка руководства пользователя системы;
- ввод в эксплуатацию.

1 АНАЛИЗ СИСТЕМЫ

1.1 Цели создания системы

Цели создания системы контроля объектов:

- сокращение временных затрат на поиск нужного объекта;
- сокращение издержек пользователей, связанных с потерей или кражей объекта;
- оптимизация рабочего времени и повышение эффективности использования оборудования и материальных ресурсов;
- выявление неэффективных звеньев в цепи производственного процесса за счет отслеживания перемещения объектов.

1.2 Задачи системы

Система дает возможность пользователям отслеживать наличие и местоположение объектов в ограниченном пространстве.

Задачи, которые должна выполнять система:

- обеспечение идентификации объектов;
- определение местоположения объектов;
- обеспечение удаленного сбора, агрегации, анализа информации, предоставляемой встроенными идентификаторами.

1.3 Характеристика объектов автоматизации и обоснование необходимости автоматизации

Предусматривается использование системы для разных категорий пользователей. Система является универсальной для единичного пользователя и для использования в промышленной отрасли.

1.3.1 Характеристика объекта автоматизации системы, используемой единичным пользователем

В этом варианте реализации системы, объектом автоматизации является жилищное помещение пользователя, такое как:

- квартира;
- частный дом;
- другое.

Использование системы в жилищном помещении обеспечит сокращение временных затрат на поиск нужного объекта и сокращение издержек пользователя, связанных с потерей или кражей объекта.

Особенности объекта автоматизации, например, такие как площадь, количество этажей, влияют на зону идентификации объектов, соответственно и на количество считывателей.

1.3.2 Характеристика объекта автоматизации системы, используемой в промышленной сфере

В промышленной сфере подразумевается автоматизация складских помещений, где нередко возникают проблемы потери груза или нехватки места для его размещения. За счет чего снижается грузооборот, и возникают простои в работе. Кроме всего прочего, персонал, работающий на складе, при получении и отгрузке товаров время от времени допускает ошибки, которые ведут к недостаче, либо излишкам. В связи с этим возникает необходимость автоматизации идентификации объектов. Внедрение системы в склад обеспечит автоматизацию инвентаризации товаров, что приведет к оптимизации рабочего времени работников склада. За счет чего также достигается уменьшение количества человеческих ресурсов, которые требуются для сбора товаров по накладным и их отгрузки.

1.4 Актуальность использования системы в жилых помещениях

Актуальность темы также подтверждается проведенным нами социальным опросом, на сайте онлайн опросников www.survio.com была составлена анкета, состоящая из следующих вопросов:

1. Пол
2. Возраст
3. Сколько сумок в Вашем гардеробе?
4. Какие вещи Вы храните в сумке?
5. Как часто Вы теряете, забываете вещи, хранящиеся в сумке? (1-никогда, 5-часто)
6. Бывали ли у Вас случаи кражи из сумки?
7. Знакомы ли Вы с RFID-технологией?
8. Имеете ли вы постоянный доступ в интернет?
9. Хотели бы Вы отслеживать наличие вещей через Интернет?

В опросе приняло участие 56 человек разных возрастных категорий, в основном женщины 66,7% рисунок (рисунок 1.1).

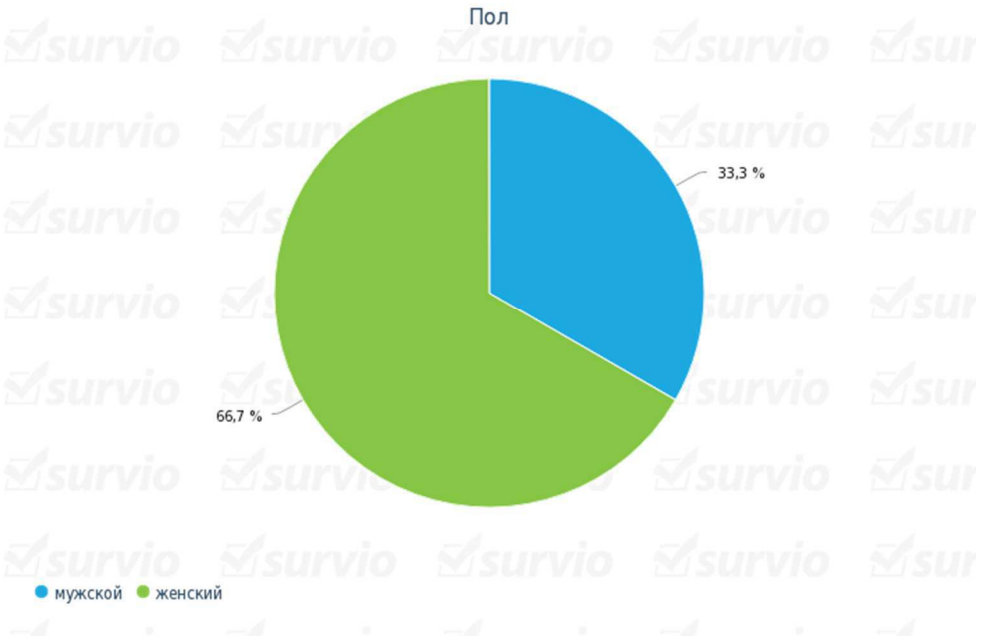


Рисунок 1.1 – Пол людей, прошедших анкетирование

В опросе приняли участие респонденты разных возрастов (смотрите рисунок 1.2).

2 Возраст

42 (2x)	21 (20x)	22 (8x)	16
20	29 (2x)	18	13
35	23 (2x)	44	55
40	38 (3x)	39	45
36	41	33	19

Рисунок 1.2 – Возрастная категория опрошенных

На вопрос «Сколько сумок в вашем гардеробе?» ответы варьировали от 2 до 5 сумок.

Рисунок 1.3 – Количество сумок в гардеробе опрошенных

На основании анализа ответов, представленного на рисунке 1.4, на вопросы под номерами 5 и 6 , можно сделать вывод, что потеря и кража вещей происходит у 1/3 опрошенных.

Рисунок 1.4 – Количество случаев кражи у опрошенных

Анализ ответов на вопрос под номером 4 представлен на рисунке 1.5.

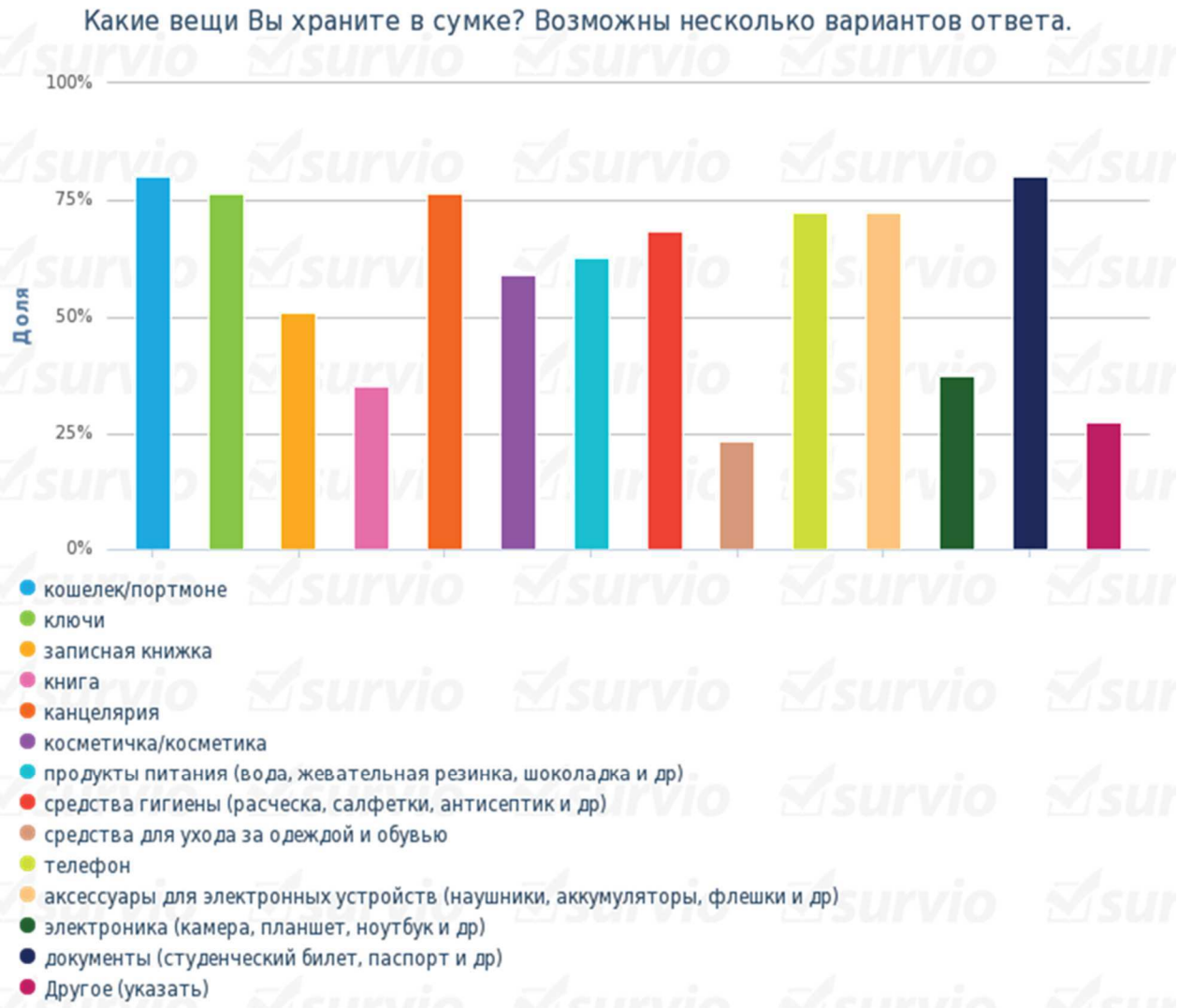


Рисунок 1.5 – Разновидности вещей, которые хранятся в сумке у опрошенных

Вопрос номер 4: «Какие вещи вы храните в сумке?» был дополнен, участниками анкетирования:

- Журналы
- Пакеты, для неожиданных покупок
- Зонт
- Сменная/спортивная одежда
- Обувь
- Лекарства
- Табак
- Очки
- Фонарь

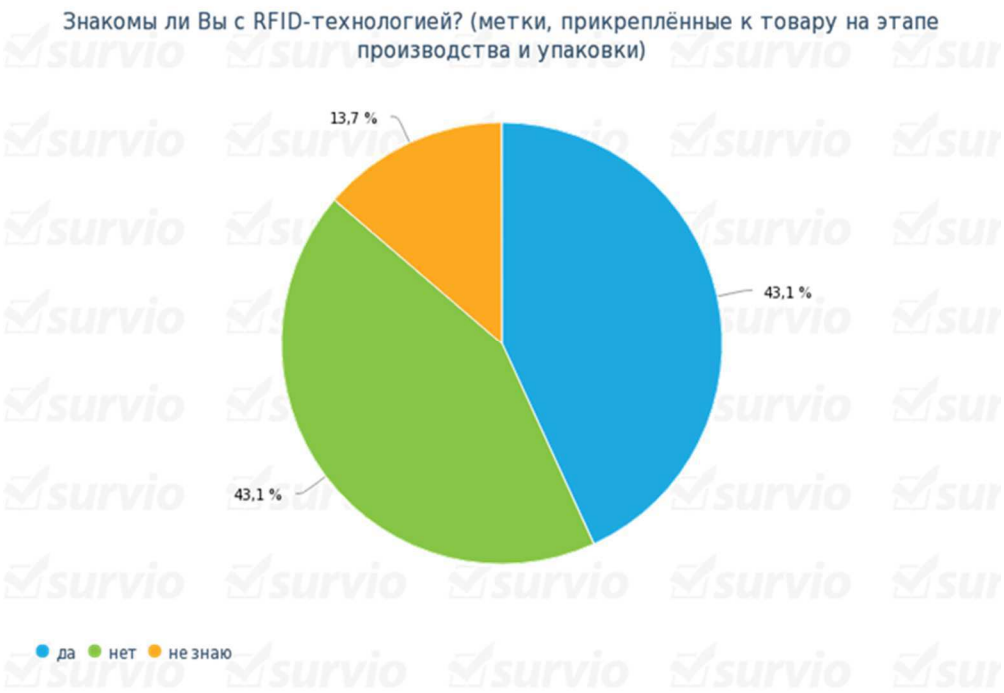


Рисунок 1.6 – Осведомленность опрошенных о технологии RFID

43,1% участников опроса знакомы с технологией RFID. Технология RFID подразумевает использование сети Интернет в качестве физической среды передачи данных, 88,2% имеют постоянный доступ в Интернет. Анализ ответов на вопрос под номером 9 представлен на рисунке 1.7.

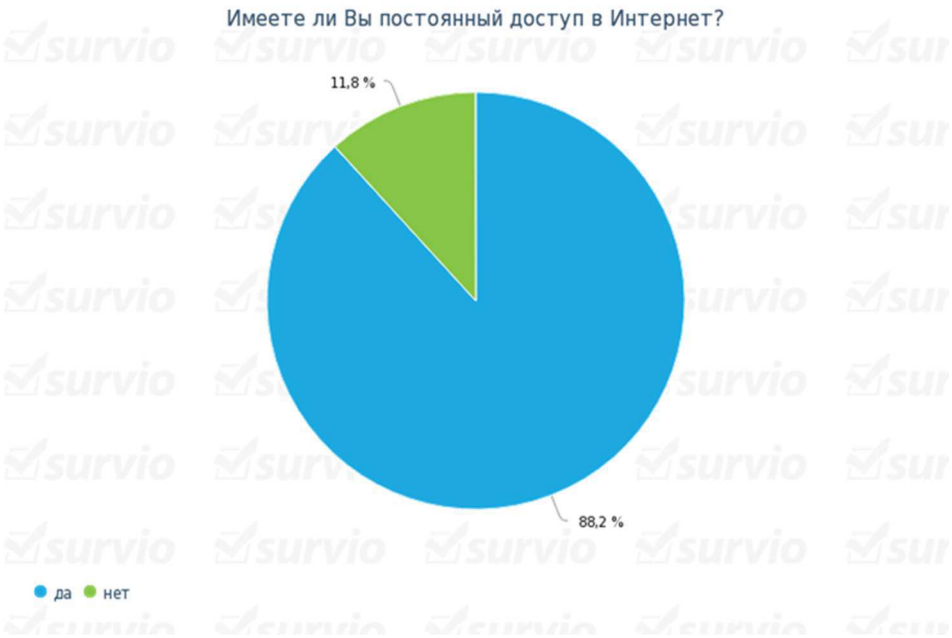


Рисунок 1.7 – Доступ опрошенных к сети Интернет

Рисунок 1.8 – Доступ опрошенных к сети Интернет

Анализ ответов на вопрос под номером 10(смотрите на рисунке 1.8) показал, что 45,1% опрошенных считают, что система отслеживания объектов посредством Интернет ресурсов, является приемлемой и заинтересованы в ней.

По итогам опроса можно сделать вывод, что система контроля объектов посредством Интернет является актуальной для разработки.

1.5 Определение требований к системе

1.5.1 Основные требования к системе, используемой в жилых помещениях

Основные требования к системе:

1. Использование сети Интернет как физической среды передач информации от идентификаторов до программного обеспечения.
2. Возможность одновременной идентификации нескольких объектов.
3. Помехоустойчивость (предотвращение возникновения коллизий при одновременной идентификации нескольких объектов).
4. Относительно небольшие размер и вес идентификаторов (не более 20 граммов).
5. Радиус действия не менее 2 метров (допустимое расстояние от идентификатора до устройства идентификации).

Возможно, возникновение новых требований в зависимости от особенностей объекта идентификации или индивидуальных пожеланий заказчика.

1.5.2 Основные требования к системе, используемой в складских помещениях

Основные требования к системе:

1. Использование сети Интернет как физической среды передач информации от идентификаторов до программного обеспечения.

2. Возможность одновременной идентификации нескольких объектов.

3. Помехоустойчивость (предотвращение возникновения коллизий при одновременной идентификации нескольких объектов).

4. Наличие основной характеристики товара:

- наименование товара;
- информация об изготовителе товара;
- цена товара;
- срок службы или срок годности товара;
- краткое описание товара.

5. Радиус действия не менее 2 метров (допустимое расстояние от индификатора до устройства идентификации).

1.6 Анализ структуры компонентов системы

Система на основе концепции «Интернет вещей» состоит из трех основных компонентов:

- объекты, оснащенные идентификаторами;
- физическая среда передачи;
- сервер.

Для выбора определенной информационной технологии необходимо провести обзор и сравнительный анализ технологий, подходящих для создания системы.

1.7 Обзор и сравнительный анализ технологий

Для выбора устройств идентификации, рассмотрим несколько технологий идентификации:

- штриховое кодирование;
- технологии GPS/ГЛОНАСС;
- Bluetooth-маячки;
- RFID (радиочастотная идентификация).

1.7.1 Штриховое кодирование (ШК).

Штриховой код, или штрих-код — технология идентификации и сбора данных, основанная на представлении информации в виде темных полосок (штрихов) и разделяющих их светлых полосок различной толщины, для последующего оптического считывания и преобразования в форму, необходимую для ее автоматического ввода в вычислительную машину. Это 13-разрядный или 8-разрядный цифровой код, представляющий собой сочетание штрихов и пробелов разной ширины. Штриховой код идентифицирует товар, потому что никакой другой товар на международном рынке не может иметь точно такой же код.

Например, цифровой 13-разрядный код товара 2005300706109 включает:

Рисунок 1.9 - Структура штрихового кода EAN-13

Для товаров небольших размеров существует структура кода EAN-8, являющаяся укороченной модификацией EAN-13. Этот код состоит из кода страны, кода изготовителя и контрольного числа (контрольное число предназначено для проверки правильности считывания кода).

Система штрихового кодирования включает в себя три компонента:

- штриховой код;
- устройство считывания;
- вычислительное устройство (компьютер).

Этапы работы штрихового кодирования:

- товару присваивается уникальный номер (товарный номер);
- товарный номер зашифровывается в виде специального символа - штрихового кода;
- штриховой код (зашифрованный товарный номер) наносится на товар;
- штриховой код автоматически считывается с товара (при выполнении логистической операции с товаром);
- штриховой код расшифровывается специальным устройством (считывателем) и предстает в цифровом виде, т.е. в виде первоначального товарного номера;
- товарный номер передается в компьютер, на котором выполняется та или иная функция управления товародвижением.

Сканер штрихового кода — это устройство, функцией которого является считывание информации с маркировки товара, декодирование и передача её в кассовый аппарат или компьютер.

По устройству считывающего элемента сканеры делятся на:

- светодиодный сканер. Принцип работы такого вида сканера заключается в использовании светодиода красного свечения в качестве источника излучения;
- лазерный сканер. Лазерный луч засвечивает штрих код, в приемник поступает отраженный световой сигнал. Далее происходит преобразование

сигнала в цифровой формат (последовательность логических нулей и единиц) и декодирование, после чего данные передаются в кассовый аппарат или компьютер.

- фотосканеры. В сканере установлена фотоголовка, которая фотографирует изображение, а затем распознает штриховой код.

Также различают такие виды сканеров штрихового кода:

- стационарные;
- ручные;
- беспроводные;
- комбинированные.

Размещают ШК на ровной поверхности упаковки товара на задней ее стенке в правом нижнем углу на расстоянии 20 мм. от краев.

Достоинства данной технологии в том, что она имеет относительно низкую стоимость и является широко распространенной технологией на сегодняшний день.

Недостатки штрихового кодирования:

- невозможно чтение без прямой видимости объекта;
- невозможна одновременная идентификация нескольких объектов;
- дальность регистрации до 4 метров.

1.7.2 Технологии GPS/ГЛОНАСС

GPS (Global Positioning System, система глобального позиционирования) — система определения местоположения объектов, основанная на использовании искусственных спутников Земли.

ГЛОНАСС (Глобальная навигационная спутниковая система) — российский аналог GPS.

Технология базируется на трех подсистемах:

- подсистемы космических аппаратов;

- подсистемы контроля и управления (ПКУ);
- подсистемы навигационной аппаратуры потребителей.

На рисунке 1.10 показано взаимодействие компонентов системы.

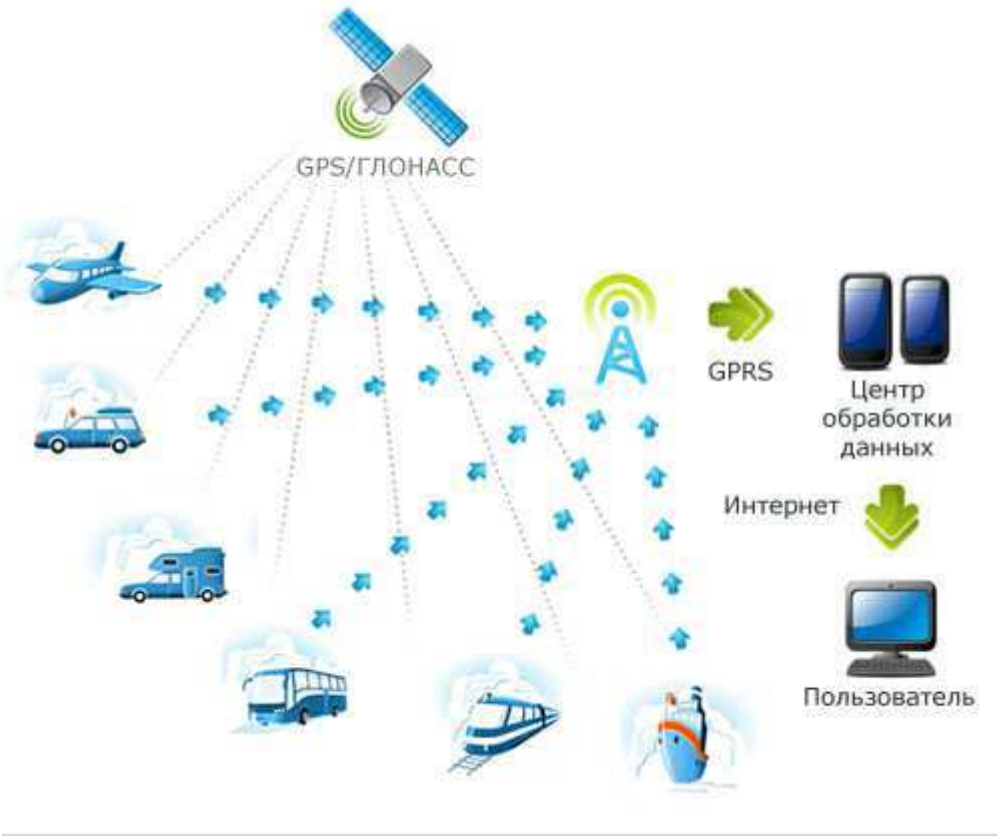


Рисунок 1.10 – Технологии GPS/ГЛОНАСС

Подсистема космических аппаратов системы состоит из 24 космических аппаратов, находящихся на круговых орбитах в трех орбитальных плоскостях. В каждой орбитальной плоскости размещаются по 8 спутников с равномерным сдвигом.

Подсистема контроля и управления состоит из Центра управления системой ГЛОНАСС и сети станций измерения, управления и контроля, рассредоточенной по всей территории России. В задачи ПКУ входят контроль правильности функционирования космических аппаратов системы ГЛОНАСС и выдача команд управления.

Навигационная аппаратура потребителей предназначена для приема навигационных сигналов спутников ГЛОНАСС и вычисления собственных координат.

Технологии GPS/ГЛОНАСС удобно использовать на открытой местности, так как большинство устройств не способны принимать спутниковый сигнал в закрытых помещениях. Более того, доступность спутникового сигнала зависит от текущей облачности, так что система будет зависеть даже от погодных условий. И ещё одним важным фактором является то, что с помощью системы GPS нельзя точно определить высоту объекта, что является препятствием на пути использования GPS в многоэтажных зданиях. Погрешность работы систем спутниковой навигации может доходить до 13 метров.

1.7.3 Bluetooth-маячки

Bluetooth - маячки представляют собой устройство, которое с заданной частотой транслирует какие-то данные, с помощью технологии Bluetooth . Маячки принципиально не могут отслеживать, они умеет только транслировать сообщения.

Рисунок 1.11 – Bluetooth-маячки

Система состоит из трех компонентов: установленные по периметру помещения маячки, включенный Bluetooth и мобильное приложение, которое уже должно быть установлено на смартфоне пользователя. Когда человек

попадает в зону действия маячка, на мобильное устройство приходит заданное для конкретного маячка push-уведомление. С помощью маячков можно организовать удобную навигацию внутри помещения. Например, владелец устройства также может проложить на карте маршрут от своего местоположения до нужного объекта на площадке. Система способна обнаружить устройство с точностью до 1 метра на расстоянии от 10 до 30 метров (в зависимости от особенностей территории).

1.7.4 RFID-технология

RFID (англ. Radio Frequency Identification, радиочастотная идентификация) — метод автоматической идентификации объектов, в основе которого лежит технология передачи данных с помощью радиоволн, необходимых для распознавания объектов, на которых размещены специальные метки, несущие как идентификационную, так и пользовательскую информацию.

Данная технология хорошо подходит для отслеживания движения некоторых объектов и получения небольшого объема информации от них. Так, например, если бы все продукты были оснащены RFID-метками, а холодильник RFID-считывателем, то он легко мог бы отслеживать срок годности продуктов, а мы могли бы, например, уходя с работы удаленно заглянуть в холодильник и определить, что нужно купить еще.

Система RFID состоит из таких компонентов как:

- RFID-метки — устройства, способные хранить и передавать данные. В памяти меток содержится их уникальный идентификационный код. Некоторые метки имеют перезаписываемую память;
- RFID-считыватель - приборы, которые читают информацию с меток и записывают в них данные. Эти устройства могут быть как постоянно подключенными к учетной системе, так и работать автономно;
- учетная система - программное обеспечение, которое накапливает и анализирует полученную с меток информацию и связывает все элементы в

единую систему. Большинство современных учетных систем (программы семейства 1С,например) уже совместимы с RFID-технологией и не требуют специальной доработки.

На рисунке 1.12 представлена RFID-система (взаимодействие компонентов).

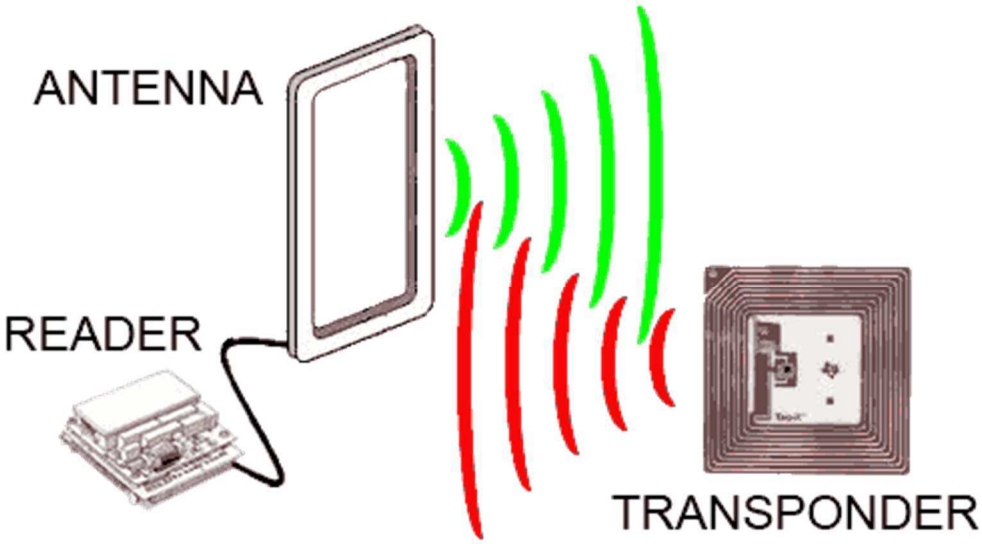


Рисунок 1.12 – RFID-система

Классификация RFID-меток:

1. По типу питания:

- активные метки — используют для передачи данных энергию встроенного элемента питания (зона чтения до 100 метров);
- пассивные метки — используют энергию, излучаемую считывателем (дальность до 8 метров).

2. По видам памяти:

- "RO" (Read Only) — данные записываются только один раз, новую информацию в них записать нельзя. Такие метки пригодны только для идентификации;
- "WORM” (Write Once Read Many) — кроме уникального идентификатора такие метки содержат блок однократно записываемой памяти, которую в дальнейшем можно многократно читать;

- "RW" (Read and Write) — такие метки содержат идентификатор и блок памяти для чтения/записи информации. Данные в них могут быть перезаписаны большое число раз.

3. По исполнению (определяется целями и условиями использования RFID-меток):

- самоклеящиеся бумажные или лавсановые метки;
- стандартные пластиковые карты;
- дисковые метки (в том числе с центральным отверстием для закрепления на палете);
- различные виды брелоков;
- специальное исполнение для жестких условий эксплуатации.

В настоящее время существует огромное многообразие меток, в зависимости от нужд заказчика, можно подобрать наиболее подходящий вариант.

Приборы для считывания данных с RFID-меток также бывают нескольких типов. По исполнению RFID-считыватели делятся на стационарные и переносные (мобильные).

1. Стационарные RFID - считыватели крепятся на стенах и потолках, могут быть выполнены в виде ворот, вмонтированы в стол или в другие места, в зависимости от нужд заказчика. Стационарные RFID - считыватели обычно напрямую подключены к компьютеру, на котором установлена программа контроля и учета. Задача таких считывателей — поэтапно фиксировать перемещение маркированных объектов.
2. Переносные RFID – считыватели обладают сравнительно меньшей дальностью действия и зачастую не имеют постоянной связи с программой контроля и учета. Мобильные считыватели имеют внутреннюю память, в которую записываются данные с прочитанных меток (потом эту информацию можно загрузить в компьютер) и, так же

как и стационарные считыватели, способны записывать данные в метку.

Чтобы считыватель RFID-системы мог взаимодействовать одновременно со многими метками, принимая при этом неискаженную информацию, необходимо учитывать ситуацию, когда несколько RFID-меток одновременно передают свои данные считывателю. А также исключить ситуацию, когда несколько RFID-считывателей пытаются считать информацию с одной метки. Такие явления называется коллизией.

Само по себе закрепление RFID - меток на объектах не способно решить проблемы учета и отслеживания. Для того чтобы построенная RFID-система эффективно решала свои задачи, она должна быть интегрирована с учетной системой. Только в том случае, если учетная программа будет полностью поддерживать функции, предоставляемые системой RFID, пользователь сможет получить максимальную прибыль от внедрения системы.

Изначально, в технологии RFID использовался диапазон низких частот (Low Frequency- LF) равный 125-134 КГц. ISO 11784 и ISO 11785 – два широко распространенных стандарта в области низких частот. С развитием данной технологии и областей ее применения, увеличилось и количество диапазонов частот, соответственно увеличилось число стандартов ISO.

В настоящее время для каждого из выделенных частотных диапазонов действуют свои стандарты, выделяются следующие диапазоны частот:

- 125-135 кГц;
- 860-930 МГц;
- 13.56 МГц;
- 2.45 ГГц.

В таблице 1.1 приведены частотные диапазоны с областью их применения.

Таблица 1.1 – Виды частотных диапазонов технологии RFID и основные области их применения.

Рабочая частота	Стандарт	Область применения
125 кГц	ISO 14223 ISO 11784	идентификация животных (в том числе домашнего скота)
13.56 МГц	ISO 14443 ISO 15693	бесконтактные смарт-карты для широкого круга приложений
860-930 МГц	ISO 15961 ISO 15962 ISO 15963 ISO 18000-6	логистика, идентификация товаров со средней дальностью
2,45 ГГц	ISO 15962 ISO 15963 ISO 18000-6 ISO 18000-4	логистика, идентификация товаров с увеличенной дальностью

Кроме широко известных стандартов ISO, широкое распространение и

популярность получили стандарты EPC Global. EPC Global выделила определенные функциональные группы меток, назвав их классами. Были выделены следующие классы:

1. Класс 0. Группа пассивных меток для идентификации объекта (Passive Identity Tag). Эти метки содержат только так называемый «электронный код продукта» (Electronic Product Code, EPC) в неизменяемом виде.
2. Класс 1. Группа пассивных меток с функциональными возможностями (Passive Functional Tag). Эта большая группа меток содержит все метки, имеющие дополнительные функции, отличающие их от первой группы. Примером таких функции могут быть перезаписываемый EPC.

3. Класс 2. Группа «полупассивных» меток (Semi-Passive Tag). К этой группе были отнесены все метки, использующие дополнительно источник питания. При этом основным источником питания должен являться считыватель, а точнее, излучаемая им энергия.
4. Класс 3. Группа активных меток (Active Tag). Эти метки содержат встроенный источник питания, полностью обеспечивающий метку необходимой энергией вне зависимости от считывателя.
5. Класс 4. Группа активных RFID-меток (RFID Tag). Эти метки не только содержат встроенный источник питания, но и набор определенной логики, позволяющей метке обмениваться данными с такой же меткой или обычным считывателем.

К достоинствам данной технологии можно отнести:

- отсутствие необходимости в прямой видимости;
- возможность идентификации нескольких объектов одновременно (до 200 меток в секунду);
- возможность считывания движущихся объектов;
- устойчивость к воздействию окружающей среды.

При работе с радиочастотной идентификацией необходимо учитывать некоторые ограничения. К ним относятся:

- относительно высокая стоимость. Стоимость пассивной радиочастотной метки составляет от 0,15 доллара (при приобретении свыше 1 000 000 шт.) до 3 долларов (при приобретении 1 шт.). В случае с метками защищенного исполнения (или на металл) эта цена может достигать 7 и более долларов;
- при размещении RFID-метки на металлическую поверхность нужно использовать метку специального типа во избежание экранирования радиосигнала;
- подверженность систем радиочастотной идентификации помехам в виде электромагнитных полей от включенного оборудования, излучающего

радиопомехи в диапазоне частот, используемом для работы RFID-системой.

Выводы по разделу один:

После обзора и сравнительного анализа вышеописанных технологий, на основании требований к системе была выбрана технология RFID.

2 ПРОЕКТИРОВАНИЕ СИСТЕМЫ

2.1 Основные этапы разработки системы

Жизненный цикл системы:

1. Предпроектное обследование (работа с заказчиком):

- обсуждение представления системы заказчиком;
- формулирование требований к системе.
- сбор и анализ материалов для проектирования;
- изучение объекта автоматизации.

2. Проектирование (разработка эскизного проекта):

- описание компонентов системы;
- выбор проектных решений разработки системы;
- планирование структуры базы данных;
- выбор комплекса технических средств (с документацией на установку).

3. Разработка:

- закупка технических средств;
- создание базы данных;
- написание программного обеспечения;
- тестирование и доводка программного комплекса;
- разработка инструкций по эксплуатации программно-технических средств (руководство пользователя).

4. Ввод в эксплуатацию:

- ввод в эксплуатацию технических средств;
- ввод в эксплуатацию программных средств;
- настройка технических и программных средств;
- обучение персонала;
- опытная эксплуатация;
- отладка и корректировка системы;
- сдача и подписание актов приемки-сдачи работ.

5. Эксплуатация:

- повседневная эксплуатация;
- общее сопровождение системы.

В разделе один было проведено предпроектное обследование, в данном разделе будет реализован этап проектирования системы.

2.2 Описание компонентов системы и выбор проектных решений разработки системы

Разрабатываемый программный комплекс будет состоять из двух частей - серверной и клиентской. Серверная часть, состоит из базы данных, в которой хранится информация обо всех объектах, подлежащих идентификации. Пользователями этой части системы являются сотрудники склада, а также пользователи системы в жилищном помещении, которые вносят в систему информацию об объектах. Клиентская часть – это приложение для взаимодействия пользователей с устройством идентификации объектов - считывателем RFID-меток.

Для написания компонентов системы контроля объектов в ограниченном объеме была выбрана интегрированная среда разработки Visual Studio 2015 и СУБД Microsoft SQL Server 2014 Express.

2.2.1 Описание компонентов системы для жилых помещений

Схема взаимодействия компонентов системы приведена ниже:

Рисунок 2.1 – Схема взаимодействия компонентов системы

2.2.1.1 Сервер

При разработке сервера предусматривается возможность хранения информации в многопользовательском режиме, поэтому для получения доступа пользователей к данным необходима установка подлинности и авторизация. Хранимые пользовательские данные не могут быть просмотрены или изменены другими людьми, таким образом достигается безопасность при хранении и пересылке данных.

Для использования системы в жилом помещении (квартира, дом) так же предусматривается привязка объектов к тому или иному пользователю. Поэтому в базе данных должна существовать такая таблица, где будет храниться информация о каждом пользователе: Фамилия, Имя, Отчество пользователя, адрес и контактный телефон, по номеру которого будет приходить SMS-оповещение при отсутствии объекта в сумке пользователя на выходе из помещения.

Сумки пользователя также будут оснащены RFID-меткой, метка должна быть активной, что поможет отследить объект (сумку) в случае кражи. Для этого в базе данных необходима таблица, где будет храниться код RFID-метки, являющейся идентификатором определенной сумки и ID пользователя сумки.

2.2.1.2 Клиентская часть

С помощью клиентской части пользователи взаимодействуют с устройством идентификации – считывателем RFID-меток. Пользователи могут добавлять новые объекты идентификации в систему, а также удалять или изменять существующие.

Для этого в базе данных должна быть таблица, где будет указываться наименование объекта и код, соответствующей объекту RFID-метки. Для внесения нового объекта в базу данных, пользователю необходимо поднести объект, оснащенный RFID-меткой, к считывателю.

2.2.2 Описание компонентов системы для складских помещений

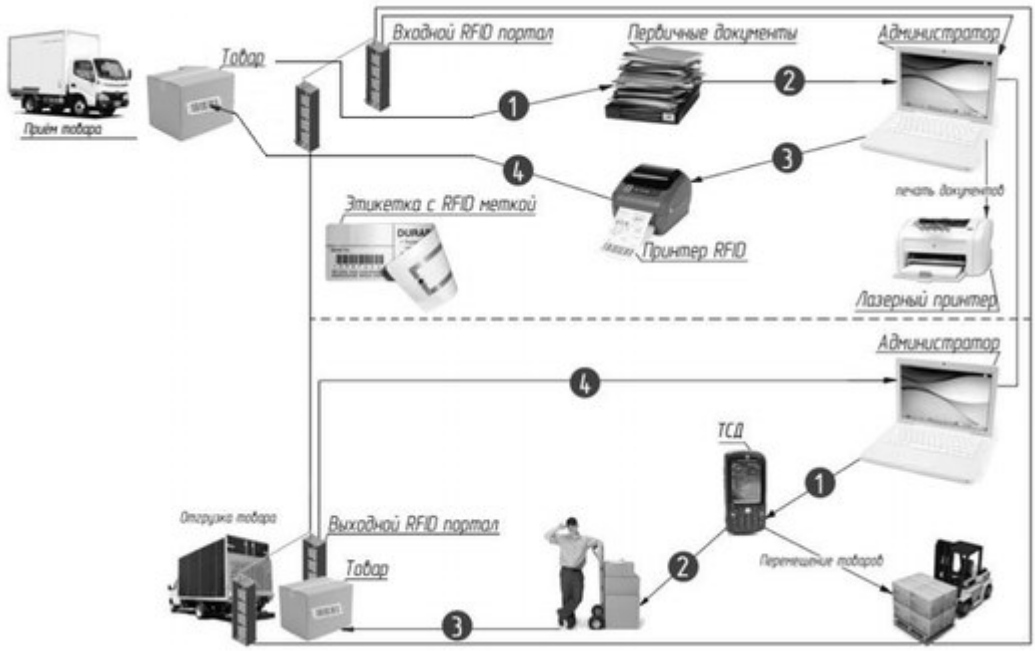


Рисунок 2.13 – Схема взаимодействия компонентов системы для склада

2.2.2.1 Сервер

В этом варианте реализации системы также предусматривается разграничение прав пользователей в зависимости от занимаемой должности. К примеру, при размещении товаров внутри склада, сотрудник, занимающийся погрузкой товаров, получает информацию о месте, куда необходимо перевезти груз. Данные о перемещениях груза заносятся в базу данных склада. По прибытии нового товара также автоматически определяется наличие свободных мест на складе. Для отслеживания наличия свободных мест на складе необходимо создать в базе данных системы, используемой на складе, такую таблицу, где товар будет связан с определенным местом хранения. Место в складе в свою очередь должно разделяться на ряды, стеллажи, полки и места на полках.

2.2.2.2 Клиентская часть

С помощью клиентской части пользователи взаимодействуют со считывателем RFID-меток. В складских помещениях может быть реализовано несколько зон идентификации объектов, соответственно может быть

установлено несколько RFID-считывателей. Каждая зона идентификации при этом будет отвечать за отдельный этап в работе склада (например, маркировка товаров, отгрузка товаров и прочее). Для этого необходимо обеспечить целостность данных всей системы, то есть, чтобы не возникало ситуаций, например, когда был отгружен товар, не прошедший этап маркировки. В этих целях необходимо написать интерфейс приложения системы таким образом, чтобы все этапы работы были разграничены, но при этом сохранялась целостность данных.

2.3 Выбор аппаратной части

2.3.1 Аппаратная часть для системы, используемой в жилых помещениях

При выборе оборудования системы для использования в жилом помещении, необходимо знать, какие именно объекты подлежат идентификации. Пользователь данного варианта использования системы может выбрать в качестве объектов как металлические, так и другие виды материалов. Из этого следует, что необходимо предусмотреть, чтобы материал, из которого сделан объект, не влиял на работу системы. Поэтому нужно выбрать несколько типов RFID-меток.

Необходимо выбрать такие составляющие системы как:

- RFID-метки;
- считыватель RFID-меток;
- рабочая частота.
- физическая среда передачи данных – беспроводная сеть Wi-Fi.

2.3.1.1 RFID-метки

Читаемость меток зависит от таких условий как:

- расстояние считываемой метки от считывателя;
- материал, из которого сделан объект, подлежащий идентификации;

Сумку пользователя целесообразно оснастить активной меткой типа WORM , предусмотрев возможность кражи или потери сумки, в ней будет записана информация о владельце сумки. Остальные объекты, находящиеся в сумке, достаточно отметить пассивной меткой типа R/O. Для объектов идентификации из металлических материалов необходимы специальные корпусные метки, устанавливаемые на металлическую поверхность.

По типу исполнения предлагается выбрать пассивные RFID-метки на клеевой основе (в виде наклейки). Для объектов из металлических материалов (например, ключей) удобнее выбрать метки в виде брелоков.

Для системы была выбрана активная метка CS3151TC.

Рисунок 2.2 –Активная метка CS3151TC

Convergence CS3151TC – это активная метка, предназначенная для применения в системах RTLS. Имеет герметичный пластиковый корпус. В стандартный комплект поставки входит метка с сенсорами движения и температуры. Технические характеристики данной метки представлены в таблице 2.1.

Таблица 2.1-Технические характеристики активной метки CS3151TC

Характеристика	Значение
Размеры	94x56x15 мм.
Вес	84 грамма
Дальность считывания	до 200 м. на открытом воздухе в условиях прямой видимости
Рабочая частота	860-960 МГц

Пассивная метка ALN-9740 'Squiggle' на клеевой основе (в виде наклейки).

Рисунок 2.3 – Пассивная метка ALN-9740 'Squiggle' на клеевой основе (в виде наклейки).

RFID-наклейка на пленочной основе. Часто применяется для интеграции неметаллических материалов, таких как пластик, бумага, синтетическая бумага, текстиль. Основные сферы применения – складская и транспортная логистика. Основные характеристики представлены в таблице 2.2.

Таблица 2.2 –Технические характеристики пассивной метки на клеевой основе (в виде наклейки)

Характеристика	Значение
Тип	многофункциональная пассивная метка EPC Class1 Gen2
Поверхность	синтетическая (пленка ПВХ)
Размеры	70 x 10 мм.
Дальность считывания	9-11 метров

Отдельный тип меток – для закрепления на металле. Обычная метка-наклейка на металлической поверхности экранируется полностью, с существенной потерей дистанции регистрации по сравнению с меткой «в воздухе». Метки «на металл» специально рассчитываются и конструируются таким образом, что их расположение на металле не снижает дистанцию регистрации, или даже увеличивает. Для металлических объектов предлагается использовать специальные RFID-метки IL-07k в виде брелоков.

Основные характеристики данной метки представлены в таблице 2.3.

Рисунок 2.4 – Пассивная метка IL-07k.

Таблица 2.3 –Технические характеристики пассивной метки IL-07k

Характеристика	Значение
Тип	многофункциональная пассивная метка EPC Class1 Gen2
Поверхность	натуральная или искусственная кожа
Размеры	39x39x5,5 мм.
Дальность считывания	2-3 метра

2.3.1.2 Считыватель

Для выбора считывателя необходимо определить его роль в системе и как именно он будет взаимодействовать с другими компонентами системы.

Считыватель необходим в данной системе для идентификации объектов пользователя на выходе пользователя из помещения. В случае отсутствия того или иного объекта, подлежащего идентификации, в сумке пользователя, система отправляет SMS-сообщение о том, что пользователь что-то забыл (в будущем предусматривается возможность разработки мобильного приложения для оповещения пользователя).

Пользователи могут добавлять новые объекты идентификации в систему, а также удалять или изменять существующие. Для этого предусматривается

покупка компьютера-моноблока со встроенным RFID-считывателем. Был выбран компьютер-моноблок TDC-5000PC с сенсорным экраном 10.4», клавиатурой и широким набором опций. Для добавления нового объекта в базу данных пользователю необходимо считать, прикрепленную на объекте, метку и ввести наименование объекта, а также указать к какой сумке относится данный объект.

Рисунок 2.5 – Компьютер-моноблок TDC-5000PC со встроенным RFID-считывателем

В таблице 2.4 описываются основные технические характеристики выбранного компьютера-моноблока.

Таблица 2.4- Основные технические характеристики TDC-5000PC

Характеристика	Значение
Размеры (ШхВхГ), мм.	264 × 211 × 52
Вес	1,5 кг.
Корпус	Пластик
Экран	10.4»
Разрешение экрана	800 × 600
Процессор	Intel Atom N270 1.6 GHz, Intel Atom D525 1.8 GHz (опция)
Оперативная память	1GB (2GB – опция)
Жесткий диск	Твердотельный накопитель 16GB (32GB – опция)
Порты и интерфейсы	2× USB, LAN, RS232, VGA, Wi-Fi (опция), Bluetooth (опция)
Клавиатура	
Доступные опции	Встроенный считыватель RFID

Продолжение таблицы 2.4

Характеристика	Значение
Питание	Поддерживается по PS/2 и USB
Интерфейсы	PS/2 – клавиатура, PS/2 – сканер штрих-кодов
Размеры (ШхВхГ)	101 × 211 × 34 мм
Доступные опции	Встроенный считыватель RFID
Питание	Поддерживается по PS/2 и USB

2.3.1.3 Рабочая частота системы

В RFID системах используются несколько рабочих частот. Каждая из них занимает определенную нишу на рынке и предназначена для решения конкретных проблем.

Низкочастотные (Low Frequency - LF) системы RFID используют диапазон частот 125-134 КГц. Обычно подобные системы используются для контроля доступа, отслеживания животных, в здравоохранении, а также в различных приложениях для платежных систем. Основные характеристики данного диапазона частот представлены в таблице 2.1.

Таблица 2.5 – Основные характеристики диапазона низких частот

Характеристика	Значение
Максимальное расстояние считывания	от 3 до 70 см.
Наличие антиколлизии	Есть, но не у всех микросхем.
Существующие типы считывателей	Стационарные, ручные считыватели.
Существующие типы радиометок	RFID- гвозди, RFID-болты, корпусные метки, брелоки, браслеты, карты.
Рекомендации по выбору меток и оборудования	Необходимо убедиться, что в списке, поддерживаемых считывателем, указан совместимый формат радиометки.

Высокочастотные (High Frequency - HF) RFID системы (13,56 МГц) работают на больших расстояниях и с большей скоростью, чем низкочастотные системы. Обычные приложения включают в себя смарт-карты, идентификацию багажа самолета и регистрации данных. Основные характеристики данного диапазона частот представлены в таблице 2.2.

Таблица 2.6 – Основные характеристики диапазона высоких частот

Характеристика	Значение
Максимальное расстояние считывания	от 3 до 100 см.
Наличие антиколлизии	Есть.
Существующие типы радиометок	Диски, брелоки, смарт-этикетки.
Существующие типы считывателей	Стационарные, ручные считыватели.
Рекомендации по выбору меток и оборудования	Требуется убедиться, что считыватель и радиометка используют один и тот же стандарт.

Ультравысокочастотные (Ultra High Frequency - UHF) системы RFID используют диапазоны частот, равные 860 – 930 МГц. Стоимость меток диапазона UHF практически равна стоимости меток диапазона HF, но при этом метки ультравысокочастотных систем обладают наибольшей дальностью считывания и обеспечивают более высокую скорость считывания.

Таблица 2.7 – Основные характеристики диапазона ультравысоких частот

Характеристика	Значение
Максимальное расстояние считывания	от 10 см до 4 м.
Наличие антиколлизии	Есть до 150 меток/сек.
Существующие типы радиометок	Корпусные метки для металлических предметов, смарт-этикетки, брелоки.

Продолжение таблицы 2.7

Характеристика	Значение
Существующие типы считывателей	Стационарные, ручные считыватели.
Рекомендации по выбору меток и оборудования	Требуется убедиться, что считыватель и радиометка используют один и тот же стандарт. Для EPC важна поддержка данного типа меток: EPC Class 0, 0+, 1,G2.

Микроволновые или сверхвысокие (SHF) системы RFID (2,45 ГГц) обеспечивают наибольшую скорость считывания, однако, при этом достаточно дорогие. Основные характеристики данного диапазона частот представлены в таблице 2.3.

Таблица 2.8 – Основные характеристики диапазона частот микроволн

Характеристика	Значение
Максимальное расстояние считывания	2-10 м.
Наличие антиколлизии	Есть до 150 меток/сек.
Существующие типы радиометок	Корпусные метки для металлических предметов.
Существующие типы считывателей	Стационарные, ручные считыватели.
Рекомендации по выбору меток и оборудования	И радиометки и считыватели должны быть от одного производителя

Расстояние считывания меток RFID-считывателем по требованиям к системе должно быть не менее 2 метров. Из вышеописанных диапазонов частот требованиям системы удовлетворяют два диапазона: UHF и SHF. В силу того, что сверхвысокие частоты отличаются большой стоимостью, для данной системы была выбрана ультравысокая частота, равная 860 – 930 МГц.

Для обеспечения беспроводной сети Wi-Fi в складе и магазине устанавливается Wi-Fi роутер ZYXEL Keenetic Lite III. Максимальная скорость беспроводного соединения300 Мбит/с.

Рисунок 2.6 - Wi-Fi роутер ZYXEL Keenetic Lite III

2.3.2 Аппаратная часть системы, используемой объектов в складских помещениях

При выборе оборудования нужно принять во внимание технические характеристики и совместимость работы системы в различных областях системы складирования товаров.

Поэтому при разработке необходимо учитывать несколько технических переменных, таких как:

- рабочая частота;
- метки;
- считыватели;
- маршрутизатор для обеспечения беспроводной связи Wi-Fi;
- принтер RFID-меток.

2.3.2.1 RFID-метки

Задача правильного выбора меток сводится к особенностям применения метки. Выбирать метки лучше исходя из используемых материалов стеллажей и конкретных реализуемых товаров. Учитывая область применения меток, будет

предложено два варианта выбора меток. Первый вариант меток будет учитывать товары, хранящиеся в материале с возможностью радиодоступа, а второй вариант для товаров в металлической поверхности.

При выборе стоит учитывать такие основные параметры метки как:

- тип метки;
- читаемость;
- объем хранения данных.

Выбирать следует из трех основных типов, это: RO, WORM или RW. Для маркировки товара целесообразнее выбрать RW метку, т.к. требуется возможность многократного считывания и записи.

Так как системе не требуются особые свойства метки (такие, как восприятие температуры, влажности и тому подобное), для корректной работы достаточно пассивных меток. Пассивные метки компактны и удобны. В отличие от активных меток, главным достоинством пассивных RFID-меток является относительно низкая стоимость. Предполагается, что цена на них будет снижаться вместе с дальнейшим развитием данной технологии.

Читаемость меток зависит от особенностей объекта автоматизации (склада).

Следует учесть такие факторы как:

- расстояние считываемой метки от считывателя;
- материал товара, к которому прикрепляется метка;
- ориентация метки в пространстве по отношению к считывателю.

Делая упор на реализацию системы в системе складирования, нужно решить, какой объем максимальный информации будет записываться на метку. Соответственно, от объема записываемой информации напрямую зависит цена метки. На метки, которые крепятся на товар, необходимой информацией является:

- наименование товара;
- информация об изготовителе товара;
- цена товара;

- срок службы или срок годности товара;
- краткое описание товара.

Выбрана самоклеящаяся RFID Метка производства МИКРОН, обладает превосходной чувствительностью и дальностью считывания, RIFD-метка отлично работает при любой ориентации относительно антенны.

Рисунок 2.7 - MIKRON M5454(2) M-3D UHF

2.3.2.2 Считыватель

Задача выбора типа считывателя состоит в правильном определении его предназначения. Основной целью его использования в системе складирования является считывание информации с меток мобильным считывателем, при маркировке товаров.

Варианты автоматизации склада:

1. RFID-считыватель на входе, метки на товарах.
2. RFID-считыватель на погрузчике, метки на товарах.
3. RFID-считыватель на погрузчике, метки на полках.
4. RFID-считыватель на ленте.
5. Создание зоны покрытия RFID-считывателями.

У каждого варианта есть плюсы и минусы, которые будут рассмотрены в таблице 2.9.

Также есть возможность совмещения вариантов, если они не противоречат друг другу. В нашей работе мы совмещаем RFID-считыватели на входе и на погрузчике, а также метки на полках и метки на товарах.

Таблица 2.9 – Варианты реализации системы на складе

Варианты автоматизации	Применимо при определенных особенностях склада	Возможность поиска свободного места	Автоматическая инвентаризация	Потребность в активных метках	Потребность в принтере	Сложность базы данных и ПО	Возможность охраны
RFID-считыватель на входе, метки на товарах	-	-	+	-	+	-	+/-
RFID-считыватель на погрузчике, метки на товарах	-	+	+	-	-	-	-
RFID-считыватель на погрузчике, метки на полках	-	+	-	-	+	+	-

Продолжение таблицы 2.9

Варианты автоматизации	Применимо при определенных особенностях склада	Возможность поиска свободного места	Автоматическая инвентаризация	Потребность в активных метках	Потребность в принтере	Сложность базы данных и ПО	Возможность охраны
RFID-считыватель на ленте	+	-	+	-	+	-	-
Создание зоны покрытия RFID-считывателями	+	+	+	+	+	+	+

Итак, необходим считыватель, закрепленный на погрузчике, для определения свободного места на складе, при размещении товаров внутри склада. А также считыватель на воротах склада, для учета товаров при отгрузке и загрузке.

Считыватель мобильный для маркировщика FEIG PRHD102-USB HF/UHF.

Рисунок 2.8 - Считыватель мобильный FEIG PRHD102-USB HF/UHF.

Ручной считыватель ID ISC.PRHD102 (устройство чтения и записи RFID-меток), снабжен антеннами двух различных диапазонов частот HF и UHF, обе антенны способны работать одновременно. Максимальная дальность считывания

меток в диапазоне HF составляет 10 см; максимальная дальность считывания меток UHF, составляет от 5 до 100 см. Сигнал об успешном чтении метки может быть подан встроенным зуммером или многоцветным светодиодом. Считыватель может использоваться как на рабочем месте, так и, например, на контроле доставки или в задачах инвентаризации, то есть можно использовать его как переносное устройство.

Таблица 2.10 - Основные технические характеристики считывателя FEIG MR101-A HF

Характеристика	Значение
Размеры, мм.	145 x 85 x 31
Вес	около 200 гр.
Корпус	Пластик
Рабочая частота	13.56 МГц

RFID-считыватель FEIG LRU3000 UHF, выбранный в качестве считывателя, устанавливаемого на ворота склада, с успехом применяется в неблагоприятных условиях, в том числе на производстве и вне помещений, для этого устройство снабжается специальным влагозащитным кожухом.

Рисунок 2.9 – Стационарный считыватель FEIG LRU3000 UHF.

Основные характеристики данного считывателя описаны в таблице 2.11.

Таблица 2.11 - Основные технические характеристики считывателя FEIG LRU3000 UHF

Характеристика	Значение
Размеры, мм.	260*157*65
Дальность считывания	12 метров
Корпус	Пластик
Рабочая частота	865 - 868 МГц

Также для взаимодействия с системой на погрузчик устанавливается компьютер-моноблок TDC-5000PC с сенсорным экраном 10.4", клавиатурой и широким набором опций.

Рисунок 2.10- Компьютер-моноблок TDC-5000PC

Таблица 2.12 - Основные технические характеристики TDC-5000PC

Характеристика	Значение
Размеры (ШхВхГ), мм.	264 × 211 × 52
Вес	1,5 кг.
Корпус	Пластик
Экран	10.4"
Разрешение экрана	800 × 600
Процессор	Intel Atom N270 1.6 GHz, Intel Atom D525 1.8 GHz (опция)
Оперативная память	1GB (2GB - опция)
Жесткий диск	Твердотельный накопитель 16GB (32GB - опция)

Продолжение таблицы 2.12

Характеристика	Значение
Порты и интерфейсы	2× USB, LAN, RS232, VGA, Wi-Fi (опция), Bluetooth (опция)
Клавиатура	
Доступные опции	Встроенный считыватель RFID
Питание	Поддерживается по PS/2 и USB
Интерфейсы	PS/2 - клавиатура, PS/2 - сканер штрих-кодов
Размеры (ШхВхГ)	101 × 211 × 34 мм
Доступные опции	Встроенный считыватель RFID
Питание	Поддерживается по PS/2 и USB

Всеми RFID-считывателями, выбранными в ходе разработки системы, поддерживается автоматический режим работы Scan-Mode и есть драйвер HID (Human Interface Device), то есть драйвер эмуляции клавиатуры. В этом случае считыватель опрашивает метки автоматически и при их обнаружении зачитывает все необходимые данные, которые будут "вставлены" в то окно программы, которое находится в фокусе и где мерцает каретка ввода данных.

Данный способ интеграции подходит практически для любой программы, где есть текстовое поле ввода данных, в том числе, способ работает даже с Excel.

2.3.2.3 Принтер RFID-меток

Для печати меток, прикрепляемых к товару, предлагается принтер Zebra RZ400, предназначенный для использования в системе складирования. Высокопроизводительные, надежные, приспособленные к жестким условиям эксплуатации RFID-принтеры серии RZ поддерживают стандарт кодировки EPC Gen2. Данные RFID-принтеры легко встраиваются в большинство

существующих складских и производственных процессов, так как RZ400 легко подключить к сети благодаря широкому выбору интерфейсов подключения: USB 2.0, беспроводной защищенный 802.11 b/g или Ethernet. Принтер удобен в эксплуатации, благодаря большому, контрастному дисплею на передней панели; легкой замене расходных материалов и простой и удобной замене печатающей головки.

Рисунок 2.11 - Принтер RFID-меток Zebra RZ400

2.3.2.4 Рабочая частота RFID-системы в складировании

Важным является выбор правильной рабочей частоты. Для этого нужно учитывать максимальное расстояния между считывателями и метками, количество меток. Предлагается выбор рабочей частоты, основанный на максимальной дальности считывания информации с метки считывателем, что необходимо при идентификации в складских помещениях.

Виды и диапазоны частот с подробной характеристикой были расписаны при выборе аппаратной части для данной системы, используемой в жилом помещении. На основании этого материала был сделан выбор рабочей частоты системы и для данного варианта, используемого в складских помещениях.

Во время процесса идентификации товара со складских стеллажей, важную роль играет скорость и расстояние считывания (расстояние считывания было выбрано равное до 4 метров). При этом наиболее подходящим частотным диапазоном является 860 - 960 МГц, который позволяет на расстоянии в

несколько метров беспрепятственно считывать информацию с большого количества меток одновременно.

Для обеспечения беспроводной сети Wi-Fi в складе устанавливается гигабитный интернет-центр с двухдиапазонной точкой доступа Wi‑Fi AC1200, управляемым коммутатором Gigabit Ethernet, портами USB 3.0 и 2.0 «Keenetic Giga III». Основные технические характеристики:

- любые тарифы до 1000 Мбит/с.;
- Wi-Fi 300 + 867 Мбит/с 2,4 + 5 ГГц с четырьмя усилителями мощности и антеннами 5 дБи;
- порты USB 3.0 и 2.0 для дисков, принтеров и модемов 3G/4G;
- файловый сервер до 40 Мбайт/с;
- защита Яндекс.DNS и SkyDNS;
- автоматическая приоритизация трафика IntelliQoS.

Рисунок 2.12 – Маршрутизатор Keenetic Giga III

Вывод по разделу два:

В качестве идентификаторов решено использовать RFID-метки трех типов:

- активные;
- пассивные - на клеевой основе (в виде наклейки);
- специальные метки, устанавливаемые на металлическую поверхность.

Активные метки нужны в качестве идентификаторов сумок пользователей системы в жилом помещении. Активная RFID-метка способна служить в качестве носителя информации, она необходима в системе в случае кражи сумки.

Пассивные RFID-метки необходимы в системе в качестве идентификаторов товаров в складских помещениях, также они служат для идентификации остальных объектов, находящихся в сумке, единичного пользователя.

Специальные метки предусматриваются для идентификации объектов с металлической поверхностью.

Рабочая частота RFID-меток и RFID-считывателя выбрана равной 860 - 960 МГц, так как именно на этой чистоте определяется необходимое расстояние считывания - до 4 метров.

Обмен между считывателем и меткой по стандарту EPC Class1 Gen2.

Эффективность применения систем Gen2 раскрывается в следующих случаях:

- одновременная регистрация до нескольких сотен меток в секунду;
- размеры зон регистрации от сантиметров до нескольких метров;
- регистрация меток в заданном направлении на расстоянии до 10 и более метров – например, меток на крупных объектах или на автотранспорте;
- регистрация метки, перемещающейся через зону регистрации мимо антенны на скорости до 250 км/ч – контроль проезда автомобилей или автопогрузчиков через ворота склада.

Современные метки стандарта Gen 2 используют эффективный механизм антиколлизий. Достигается он путем генерации в начале запроса считывателем и меткой числа q (в диапазоне от 0 до 2) в степени n. Обмен информацией происходит при совпадении числа q считывателя и метки. В случае, когда это число q нескольких меток совпало с числом считывателя, производится новый запрос и q генерируется заново. Если же такая ситуация возникает часто, считыватель корректирует степень двойки n, изменяя границы диапазона.

Считыватели RFID-меток также выбраны нескольких видов:

- компьютер-моноблок, со встроенным RFID-считывателем, для идентификации объектов в жилом помещении;
- стационарный RFID-считыватель считыватель (монтируемый в ворота) для системы идентификации в складе;
- мобильный считыватель для маркировки товаров в складе;
- компьютер-моноблок, со встроенным RFID-считывателем, (на погрузчик) для нахождения свободных мест в складе.

3 РАЗРАБОТКА СИСТЕМЫ

3.1 Создание базы данных системы

3.1.1 Создание базы данных системы идентификации объектов в жилых помещениях (зависит от особенностей объекта автоматизации и индивидуальных требований заказчика)

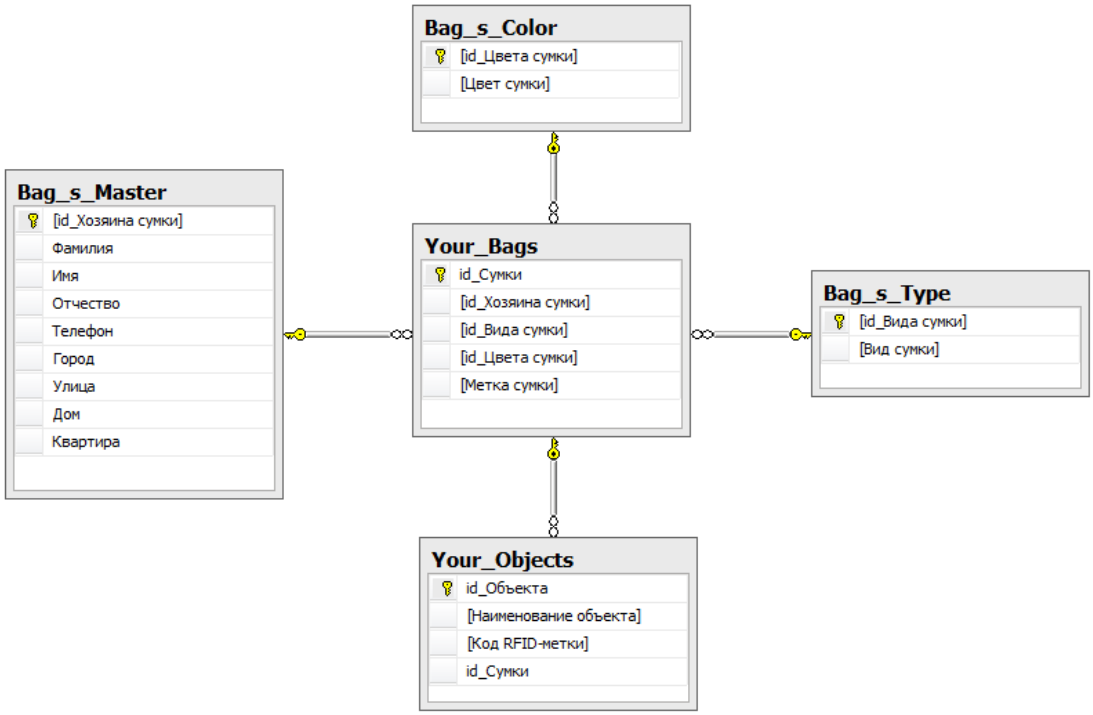


Рисунок 3.1 –Диаграмма базы данных системы для жилых помещений

Описание таблиц

Таблица Your_Objects хранит данные об объектах идентификации, находящихся в сумке пользователя.

Таблица Your_Objects включает в себя:

1. id_Объекта – первичный ключ базы данных;
2. Наименование объекта – атрибут, служащий для хранения наименования объекта идентификации;
3. Код RFID-метки – атрибут, служащий для хранения кода RFID-метки, прикрепленной к объекту идентификации;
4. id_Сумки – первичный ключ базы данных для связи с таблицей Your_Bags.

Имя столбца	Тип данных	Разрешит...
id_Объекта	nchar(10)	☐
[Наименование объе...	nchar(20)	☐
[Код RFID-метки]	nchar(14)	☐
id_Сумки	nchar(10)	☐
		☐

Рисунок 3.2 – Проект таблицы Your_Objects

Таблица Bag_s_Master хранит данные о пользователях системы, которым принадлежат объекты идентификации.

Таблица Bag_s_Master включает в себя:

1. id_Хозяина сумки – первичный ключ базы данных;
2. Фамилия – атрибут, служащий для хранения фамилии пользователя;
3. Имя - атрибут, служащий для хранения имени пользователя;
4. Отчество – атрибут, служащий для хранения имени пользователя;
5. Телефон – атрибут, служащий для хранения номера телефона пользователя, на который приходит SMS-сообщение, когда пользователь забывает взять с собой объект идентификации;
6. Город – атрибут, служащий для хранения названия города, в котором проживает пользователь;
7. Улица – атрибут, служащий для хранения названия улицы, на которой проживает пользователь;
8. Дом – атрибут, служащий для хранения номера дома, в котором проживает пользователь;
9. Квартира – атрибут, служащий для хранения номера квартиры, в которой проживает пользователь.

Имя столбца	Тип данных	Разрешит...
[id_Хозяина сумки]	nchar(10)	
Фамилия	nchar(20)	
Имя	nchar(20)	
Отчество	nchar(20)	
Телефон	float	
Город	nchar(20)	
Улица	nchar(20)	
Дом	nchar(10)	
Квартира	nchar(10)	

Рисунок 3.3 – Проект таблицы Bag_s_Master

Таблица Your_Bags хранит данные о сумках пользователя.

Таблица Your_Bags включает в себя:

1. id_Сумки - первичный ключ базы данных;
2. id_Хозяина сумки – первичный ключ базы данных для связи с таблицей Bag_s_Master;
3. id_Вида сумки - первичный ключ базы данных для связи с таблицей Bag_s_Type;
4. id_Цвета сумки – первичный ключ базы данных для связи с таблицей Bag_s_Color;
5. Метка сумки – атрибут,служащий для хранения кода RFID-метки сумки.

Имя столбца	Тип данных	Разрешит...
id_Сумки	nchar(10)	
[id_Хозяина сумки]	nchar(10)	
[id_Вида сумки]	nchar(10)	
[id_Цвета сумки]	nchar(10)	
[Метка сумки]	nchar(14)	

Рисунок 3.4 – Проект таблицы Your_Bags

Таблица Bag_s_Type хранит данные о видах сумок пользователя.

Таблица Bag_s_Type включает в себя:

1. id_Вида сумки – первичный ключ базы данных;
2. Вид сумки – атрибут, служащий для хранения вида сумки пользователя;

Имя столбца	Тип данных	Разрешит...
[id_Вида сумки]	nchar(10)	
[Вид сумки]	nchar(10)	

Рисунок 3.5 – Проект таблицы Bag_s_Type

Таблица Bag_s_Color хранит данные о расцветках сумок пользователя. Таблица Bag_s_Color включает в себя:

1. id_Цвета сумки – первичный ключ базы данных;
2. Цвет сумки – атрибут, служащий для хранения цветов сумок пользователя.

Имя столбца	Тип данных	Разрешит...
[id_Цвета сумки]	nchar(10)	
[Цвет сумки]	nchar(20)	

Рисунок 3.6 – Проект таблицы Bag_s_Color

3.1.2 Создание базы данных системы идентификации объектов в складских помещениях

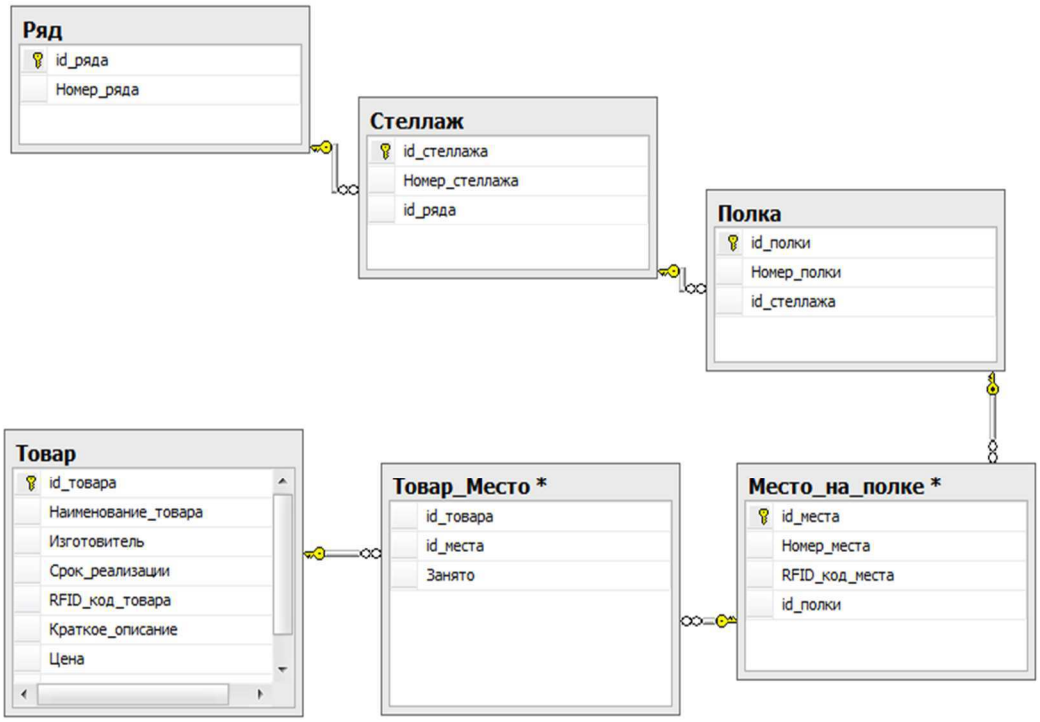


Рисунок 3.7-Диаграмма базы данных системы для склада

Описание таблиц

Таблица Товар хранит в себе данные о товарах на складе.

Таблица Товар включает в себя:

1. id_товара-первичный ключ базы данных;
2. Наименование_товара – атрибут, служащий для хранения наименование товаров, являющихся объектами идентификации;
3. Изготовитель – атрибут, служащий для хранения информации об изготовителе товара;
4. Срок_реализации – атрибут, служащий для хранения информации о сроках реализации товара;
5. RFID-код_товара – атрибут, служащий для хранения кода идентификации RFID-метки, прикрепленной к товару;
6. Краткое_описание – атрибут, служащий для хранения информации, кратко описывающей товар;
7. Цена – атрибут, служащий для хранения информации о стоимости товара;
8. На_складе – атрибут, служащий для хранения информации о наличии или отсутствии товара на складе.

Имя столбца	Тип данных	Разрешит...
id_товара	nchar(10)	☐
Наименование_товара	nchar(20)	☐
Изготовитель	nchar(20)	☐
Срок_реализации	nchar(10)	☑
RFID_код_товара	nchar(14)	☐
Краткое_описание	nchar(50)	☐
Цена	int	☐
На_складе	bit	☐
		☐

Рисунок 3.8 - Проект таблицы Товар

Таблица Место_на_полке хранит информацию о местах на полке, где находится тот или иной товар.

Таблица Место_на_полке включает в себя:

1. id_Места - первичный ключ базы данных;
2. Номер_места – атрибут, служащий для хранения информации о номере места в ряде, где хранится товар;
3. RFID_код_места – атрибут, служащий для хранения кода идентификации RFID-метки места;
4. id-полки - первичный ключ базы данных для связи с таблицей Полка;

Имя столбца	Тип данных	Разрешит...
id_места	nchar(10)	
Номер_места	int	
RFID_код_места	nchar(14)	
id_полки	nchar(10)	

Рисунок 3.9 - Проект таблицы Место_на_полке

Таблица Товар_Место является промежуточной для таблиц Товар и Место_на_полке.

Таблица Товар_Место включает:

1. id_товара – первичный ключ базы данных для связи с таблицей Товар;
2. id_места – первичный ключ базы данных для связи с таблицей Место_на_полке.
3. Занято – атрибут, служащий для хранения информации о том, занято то или иное место или свободно.

Имя столбца	Тип данных	Разрешит...
id_товара	nchar(10)	
id_места	nchar(10)	
Занято	bit	

Рисунок 3.10 - Проект таблицы Товар_Место

Таблица Полка хранит информацию о полках на складе, на которых хранятся объекты, подлежащие идентификации.

Таблица Полка включает в себя:

1. id_полки – первичный ключ базы данных;
2. номер_полки – атрибут, служащий для хранения информации о номере полки;
3. id_стеллажа – первичный ключ базы данных для связи с таблицей Стеллаж.

Имя столбца	Тип данных	Разрешит...
id_полки	nchar(10)	
Номер_полки	int	
id_стеллажа	nchar(10)	

Рисунок 3.11 - Проект таблицы Полка

Таблица Стеллаж хранит информацию о стеллажах на складе, на которых размещаются объекты, подлежащие идентификации.

Таблица Стеллаж включает в себя:

1. id_стеллажа – первичный ключ базы данных;
2. номер_стеллажа – атрибут, служащий для хранения номера стеллажа, на котором размещается товар;
3. id_ряда – первичный ключ базы данных для связи с таблицей Ряд.

Имя столбца	Тип данных	Разрешит...
id_стеллажа	nchar(10)	
Номер_стеллажа	int	
id_ряда	nchar(10)	

Рисунок 3.12 - Проект таблицы Стеллаж

Таблица Ряд хранит информацию о рядах, на которых размещается товар на складе.

Таблица Ряд включает в себя:

1. id_ряда - первичный ключ базы данных;
2. номер_ ряда – атрибут, служащий для хранения номер ряда, на котором размещается товар на складе.

Имя столбца	Тип данных	Разрешит...
id_ряда	nchar(10)	☐
Номер_ряда	int	☐
		☐

Рисунок 3.12 - Проект таблицы Ряд

3.2 UML-диаграмма использования системы

3.2.1 UML-диаграмма использования системы в жилых помещениях

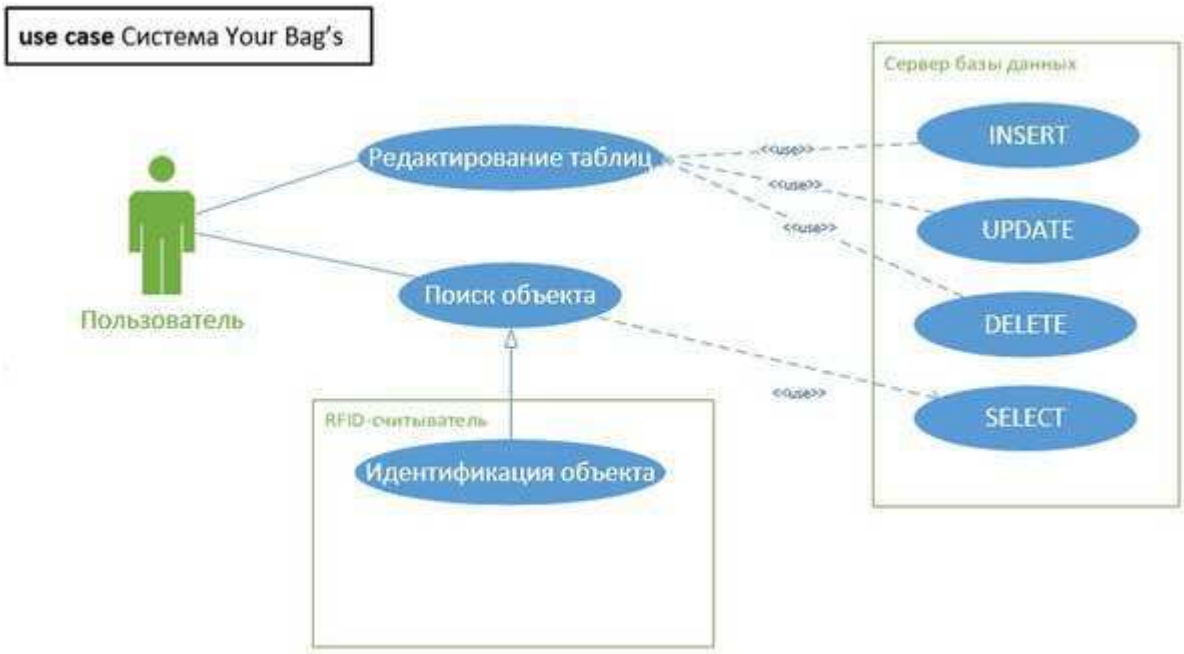


Рисунок 3.13 - UML-диаграмма использования системы для жилых помещений

3.2.2 UML-диаграмма использования системы в складских помещениях

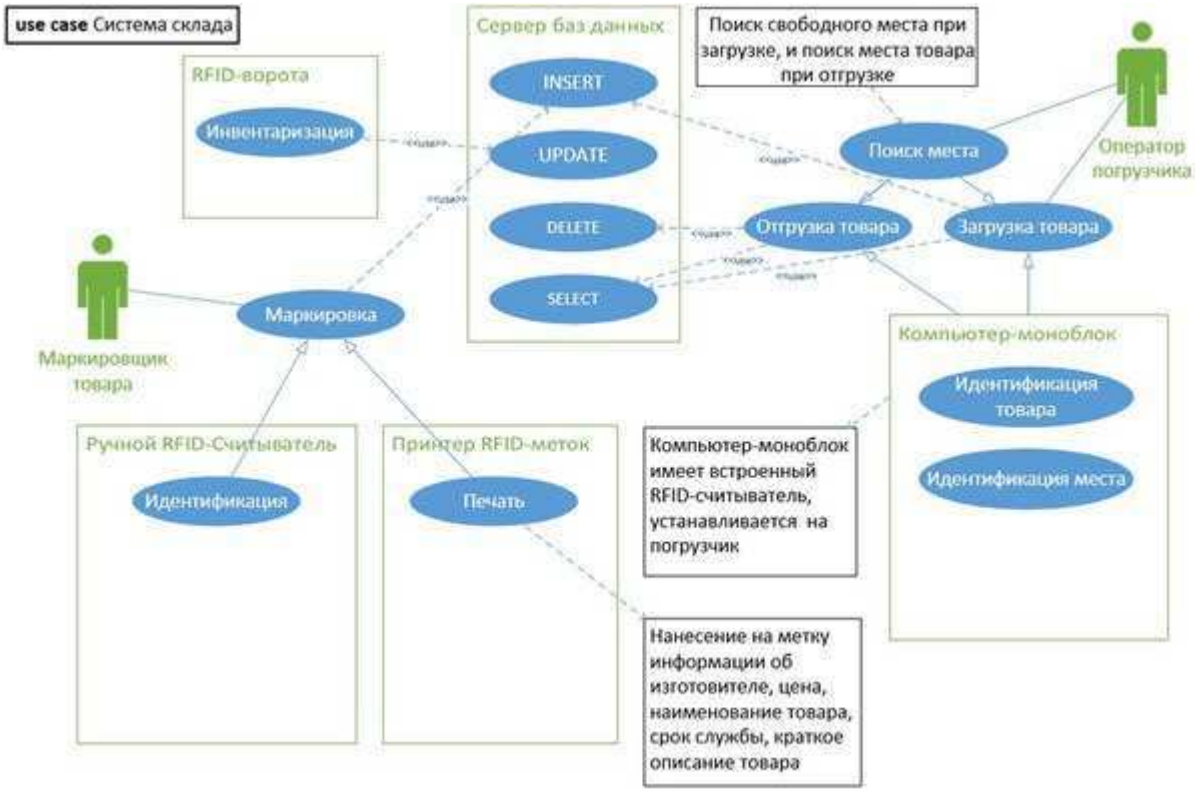


Рисунок 3.14 - UML-диаграмма использования системы для складских помещений

3.3 Разработка интерфейса приложения системы

3.3.1 Разработка интерфейса приложения системы, используемой в жилых помещениях

Программа, взаимодействующая с базой данных и RFID-считывателем, написана на языке C# .NET с использованием Windows Forms и выглядит, как показано на рисунках 3.15 и 3.16.

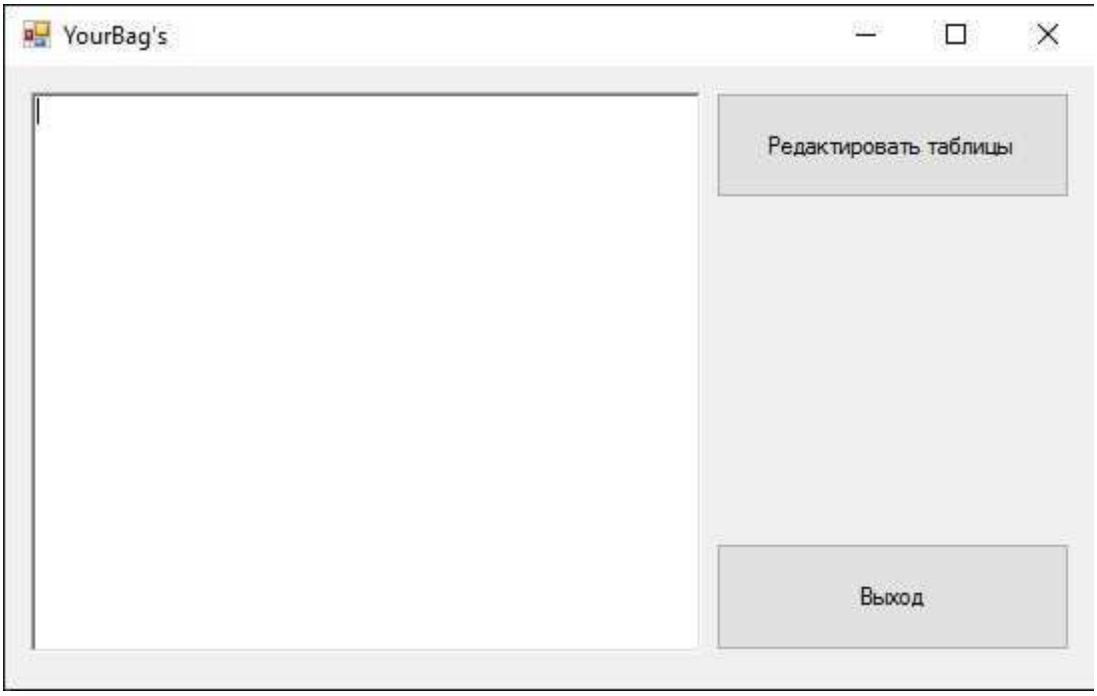
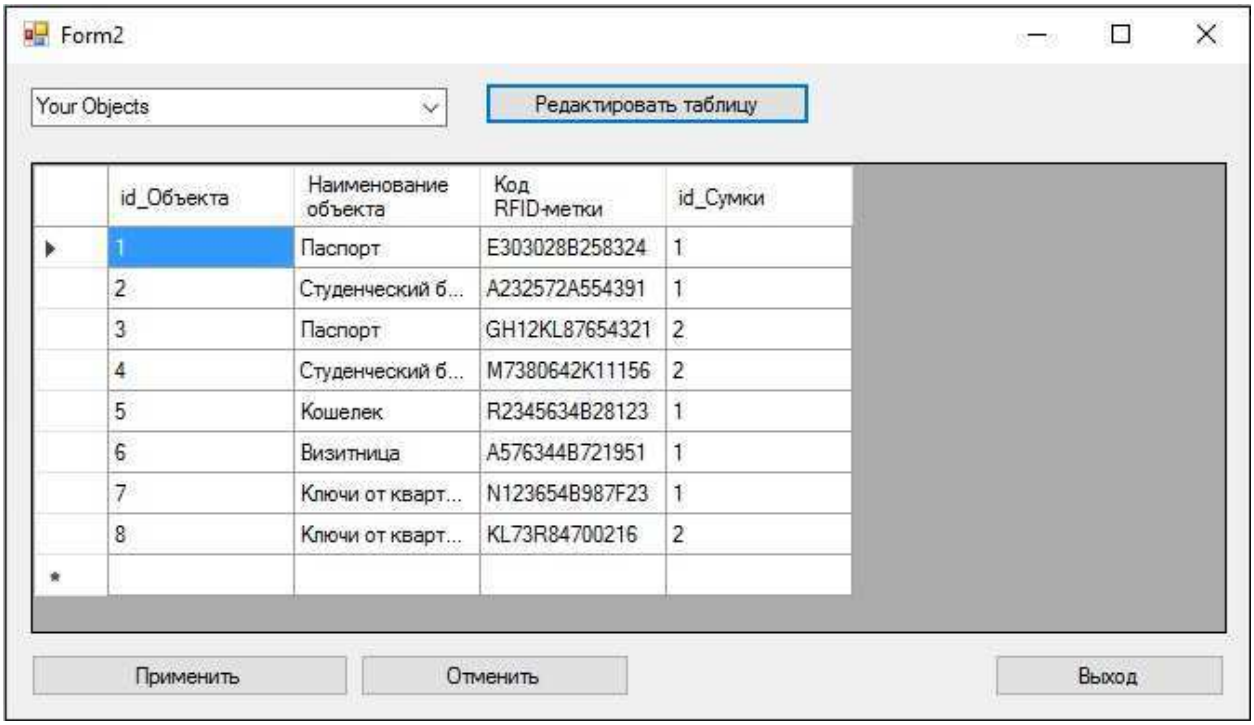


Рисунок 3.15 - Главная форма программы



id_Объекта	Наименование объекта	Код RFID-метки	id_Сумки
1	Паспорт	E303028B258324	1
2	Студенческий б...	A232572A554391	1
3	Паспорт	GH12KL87654321	2
4	Студенческий б...	M7380642K11156	2
5	Кошелек	R2345634B28123	1
6	Визитница	A576344B721951	1
7	Ключи от кварт...	N123654B987F23	1
8	Ключи от кварт...	KL73R84700216	2

Рисунок 3.16 - Форма редактирования таблиц

Подключение к базе данных настраивается с помощью строки подключения: connection.ConnectionString=(@"Data Source=LENOVOG500\SQLEXPRESS;Integrated Security=SSPI;Persist Security Info=False;Initial Catalog=Your_Bags");

RFID-считыватель, по средствам драйвера HID (Human Interface Device) взаимодействует с элементом главной формы richtextbox. RFID-коды сканированных меток появляются в поле, сравниваются с кодами сумок, после чего программа запоминает ID сумки и сравнивает оставшиеся метки с метками объектов в данной сумке.

Если количество просканированных меток не совпадает с количеством объектов в сумке, программа берёт из базы данных названия забытых элементов и формирует SMS-сообщение: «Вы забыли: наименование предмета». Формируется запрос и отправляться на SMS-шлюз посредствам Интернета. SMS-шлюз выполняет доставку сообщения на абонентский номер хозяина сумки.

Редактирование, добавление и удаление записей в базу данных происходит посредствам специальной формы. Для редактирования следует выбрать таблицу в выпадающем меню наверху и нажать кнопку «Редактировать таблицу». Далее внести изменения в поле с таблицей и нажать кнопку «Применить» для сохранения, или «Отменить» для отмены изменений.

3.3.2 Разработка интерфейса приложения системы, используемой в складских помещениях

Так как на складе используется три RFID-считывателя, программное обеспечение состоит из трех продуктов:

1. Программа для ручного считывателя маркировщика прибывшего товара.

Программа, взаимодействующая с базой данных и RFID-считывателем, написана на языке C# .NET с использованием Windows Forms и выглядит, как показано на рисунке 3.17.

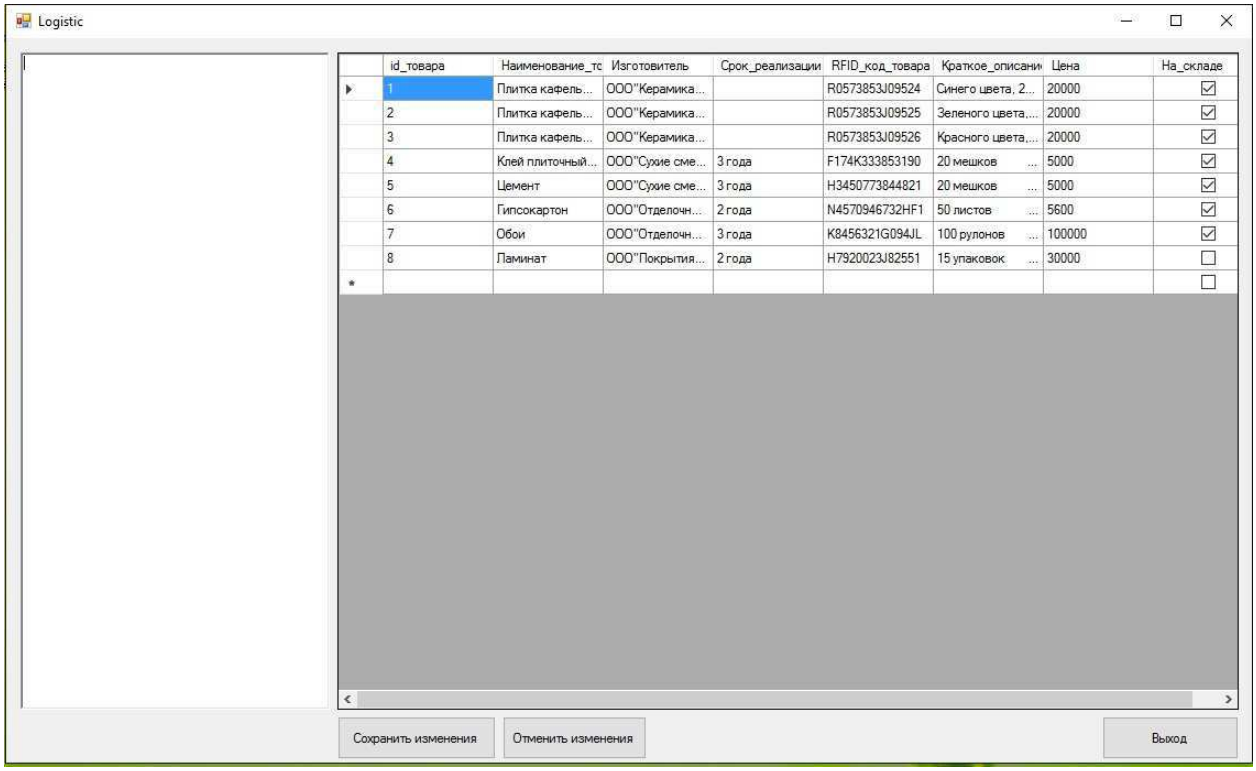


Рисунок 3.17 - Программа маркировщика товаров

Подключение к базе данных настраивается с помощью строки подключения: connection.ConnectionString=(@"Data Source=LENOVOG500\SQLEXPRESS;Integrated Security=SSPI;Persist Security Info=False;Initial Catalog=Storage");

RFID-считыватель, по средствам драйвера HID (Human Interface Device) взаимодействует с элементом формы richtextbox. RFID-коды сканированных меток появляются в поле.

После сканирования меток, маркировщик заполняет базу, копируя RFID-коды в соответствующее поле базы данных. При этом поле «На_складе» должно оставаться незаполненным. После внесения всех записей следует нажать кнопку «Сохранить изменения».

2. Программа для RFID-ворот на складе.

Программа, взаимодействующая с базой данных и RFID-считывателем, написана на языке C# .NET с использованием Windows Forms и выглядит, как показано на рисунке 3.15.

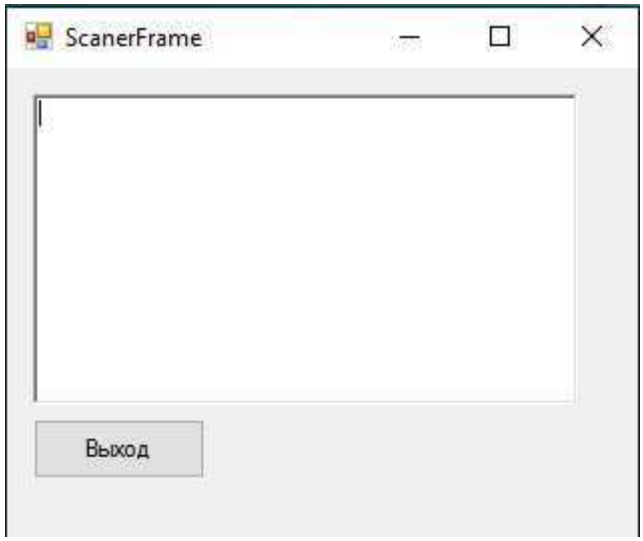


Рисунок 3.18 - Программа для RFID-ворот

Подключение к базе данных настраивается с помощью строки подключения:

connection.ConnectionString=(@"Data Source=LENOVOG500\SQLEXPRESS;Integrated Security=SSPI;Persist Security Info=False;Initial Catalog=Storage");

RFID-считыватель, по средствам драйвера HID (Human Interface Device) взаимодействует с элементом формы richtextbox. RFID-коды сканированных меток появляются в поле. После чего, программа сверяет коды с базой данных и устанавливает галочку в поле «На_складе» при въезде, и снимает её при выезде.

```
string sqlfalse = string.Format("Update Товар Set На_складе = '{0}' Where id_товара = '{1}'", 0, idObj);
string sqltrue = string.Format("Update Товар Set На_складе = '{0}' Where id_товара = '{1}'", 1, idObj);
        if (Checking == "True")
        {
            using (SqlCommand cmd = new SqlCommand(sqlfalse, connection))
         {cmd.ExecuteNonQuery() }
        }
        if (Checking == "False")
       {
            using (SqlCommand cmd = new SqlCommand(sqltrue, connection))
```

```
        { cmd.ExecuteNonQuery();}
    }
```

3. Программа для компьютера-моноблока на погрузчике.

Программа, взаимодействующая с базой данных и RFID-считывателем, написана на языке C# .NET с использованием Windows Forms и выглядит, как показано на рисунках 3.16, 3.17 и 3.18.

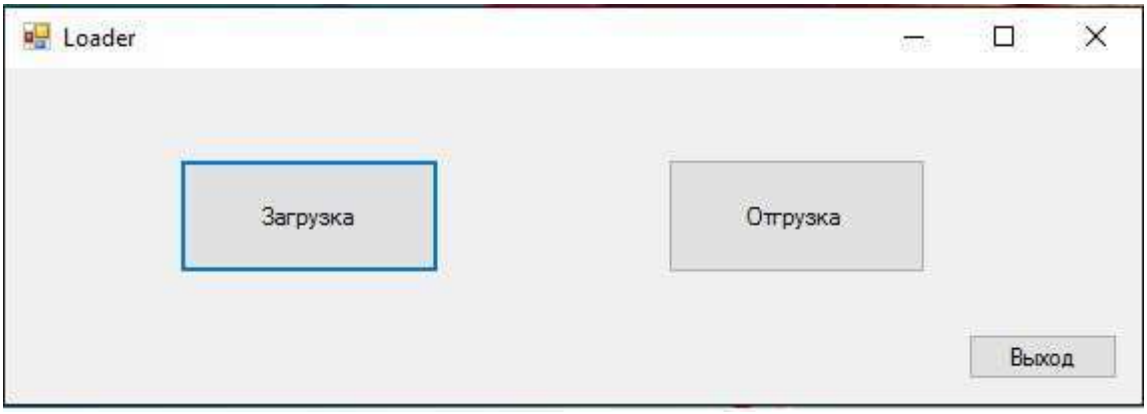


Рисунок 3.19 - Главная форма программы погрузчика

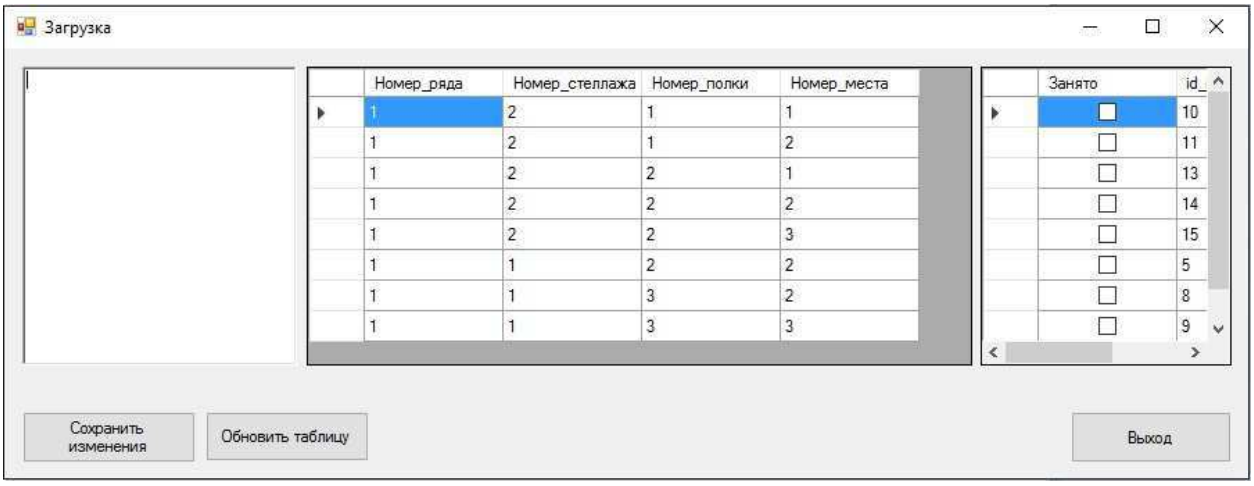


Рисунок 3.20 - Форма для загрузки товара на склад

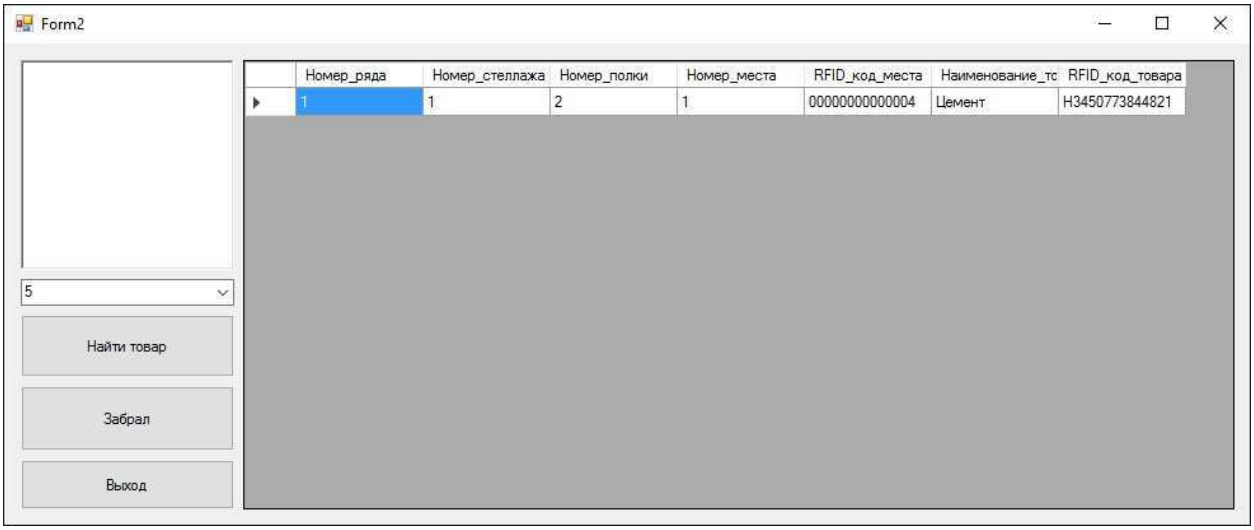


Рисунок 3.21 - Форма для отгрузки товара со склада

Подключение к базе данных настраивается с помощью строки подключения: connection.ConnectionString=(@"Data Source=LENOVOG500\SQLEXPRESS;Integrated Security=SSPI;Persist Security Info=False;Initial Catalog=Storage");

При загрузке товара на склад, оператор погрузчика выбирает форму «Загрузка». На форме отображаются свободные места на складе. Оператор выбирает свободное место и ставит товар на нужное место на полке.

В этот момент, RFID-считыватель, по средствам драйвера HID (Human Interface Device) взаимодействует с элементом формы richtextbox. Оператор погрузчика ставит отметку напротив выбранного места и нажимает кнопку «Сохранить изменения». Программа устанавливает отметку «Занято» записи места и связывает место с товаром.

При отгрузке товара со склада, оператор погрузчика выбирает форму «Отгрузка». На ней он выбирает ID товара, указанный в накладной, и получает информацию о месте расположения на складе и код метки.

RFID-считыватель, по средствам драйвера HID (Human Interface Device) взаимодействует с элементом формы richtextbox. Оператор сравнивает код товара и код места из базы, с кодами, полученными со считывателя, и, при совпадении, нажимает кнопку «Забрал». Программа снимает отметку «Занято» с

места и удаляет связь между местом и товаром. После чего, товар может покинуть склад.

Вывод по разделу три:

В данной главе была разработана база данных системы и написано программное обеспечение.

4 РУКОВОДСТВО ПОЛЬЗОВАТЕЛЯ СИСТЕМЫ

4.1 Руководство пользователя системы, используемой в жилых помещениях

4.1.1 Условия эксплуатации

Условия эксплуатации системы классифицируются по трем основным критериям:

- радиочастотные препятствия. Радиочастотными препятствиями могут являться металлические изделия, такие как металлические опоры, рамы, которые отражают и блокируют прохождение радиоволн от считывателя к меткам;
- условия окружающей среды. Рассматривая данный пункт, необходимо принимать во внимание влажность и температуру воздуха, а также статическое электричество. Данные факторы влияют на срок службы метки, а также, избыточная влажность воздуха может ослаблять радиосигналы.
- радиочастотная интерференция. Здесь необходимо учитывать расположение и параметры Wi-Fi-роутера. Данные устройства могут генерировать радиопомехи, которые взаимодействуют с RFID.

Принимая во внимание вышеприведенные факторы, нужно устанавливать оборудование таким образом, чтобы посторонние объекты не сказывались на функциональности RFID-системы. Если в конструкции здания присутствуют металлические изделия, то оборудование нужно по возможности отдалять от них. Так же, необходимо принимать меры по регулированию влажности воздуха.

4.1.2 Интеграция с существующими системами

Так как система используется в жилищном помещении, под возможность интеграции попадает только сеть Интернет. Если в жилищном помещении присутствует Wi-Fi сеть, нет необходимости в её создании.

4.1.3 Обслуживание

При разработке RFID-системы необходимо планировать дополнительные для нее ресурсы заранее. Для этого существуют необходимые параметры:

- замена поврежденного оборудования. Нельзя исключать возможности выхода из строя любой единицы оборудования;
- техническая поддержка. Сюда необходимо отнести ресурсы технической поддержки: договоры на поддержку;
- усовершенствование. Индивидуальные настройки для поддержки новых или расширения существующих функций.

4.1.4 Эксплуатация

RFID-считыватель не должен отключаться от сети питания, а также подвергаться вредному внешнему воздействию (ударам, избыточной влажности, влиянию статического электричества).

RFID-метки крепятся к объектам в зависимости от их особенностей. Для ключей используется специальная метка для металлических поверхностей.

4.2 Руководство пользователя системы, используемой в складских помещениях

4.2.1 Условия эксплуатации

Условия эксплуатации RFID-системы можно классифицировать по трем критериям:

- радиочастотные препятствия. Радиочастотные препятствия могут возникать вследствие присутствия радионепрозрачных объектов и мобильных устройств, которые эффективно блокируют прохождение радиоволн от антенн считывателя к меткам. Так же, в конструкции склада могут быть металлические изделия, такие как металлические

опоры, рамы, которые отражают радиоволны и оказывают похожее влияние;

- условия окружающей среды. Рассматривая данный пункт, необходимо учитывать влажность воздуха на складе, присутствие статического электричества, температуры воздуха, ударов и вибрации. Данные факторы при их избытке отрицательным образом сказываются на сроке службы метки. Избыточная влажность воздуха может ослаблять радиосигналы и приводить к недостаточности энергии, требуемой для считывания информации с метки;
- радиочастотная интерференция. Здесь необходимо учитывать параметры таких устройств, как Bluetooth, Wi-Fi, беспроводные рации, электродвигатели. Данные устройства могут генерировать радиопомехи, которые взаимодействуют с RFID-системой. Принимая во внимание вышеприведенные факторы, нужно устанавливать оборудование таким образом, чтобы посторонние объектов не сказывались на функциональности RFID-системы. Если в конструкции здания присутствуют металлические изделия, то оборудование нужно по возможности отдалять от них. Так же, необходимо принимать меры по регулированию влажности воздуха.

4.2.2 Интеграция с существующими системами

Интеграция RFID-системы с существующим оборудованием может благотворно сказываться на функциональности работы всей системы в целом. Принимая во внимание то, что в настоящее время существует большое множество различных современных систем, их можно легко интегрировать в реализуемую RFID-систему.

4.2.3 Обслуживание

При разработке RFID-системы необходимо планировать дополнительные для нее ресурсы заранее. Для этого существуют необходимые параметры:

- замена поврежденного оборудования. Нельзя исключать возможности выхода из строя любой единицы оборудования. Это могут быть считыватели, метки, принтеры;
- лицензии на оборудование и микропрограммы и их модернизация. Данный пункт включает в себя лицензии и модернизацию считывателей, принтеров, ПК и другое;
- персонал. Данный пункт предусматривает сотрудников и консультантов для обслуживания RFID-системы;
- техническая поддержка. Сюда необходимо отнести ресурсы для создания среды технической поддержки, контракты на поддержку с изготовителями;
- усовершенствование. Индивидуальные настройки RFID-системы для поддержки новых или расширения существующих функций, новые интерфейсы.

4.2.4 Эксплуатация

Маркировка товара RFID-метками должна происходить до попадания груза на склад, в разгрузочной зоне. Следует учитывать особенности маркируемой поверхности и выбирать метку, исходя из материала и конструкции товара.

Груз, проходящий через RFID-ворота, должен быть оснащен меткой и внесен в базу.

При внесении товара в базу, поле, отвечающее за наличие товара на складе, должно остаться пустым, так как оно заполняется программным обеспечением, при прохождении через RFID-ворота.

Оператор погрузчика при ввозе товара на склад работает с формой «Загрузка», где выбирает свободное место на складе из предложенных системой. Когда товар окажется на нужном месте, ставит отметку напротив записи места, на которое он выгрузил товар, и нажимает кнопку «Сохранить изменения».

При отгрузке товара оператор погрузчика получает накладную на нужные товары, выбирает форму «Отгрузка», находит нужный груз и получает сведения о местоположении товара. При подъеме товара с полки, оператор должен нажать кнопку «Забрал», для удаления привязки груза к месту. После чего груз может покинуть склад.

Вывод по разделу 4:

В данной главе было разработано руководство пользователя системой в двух вариантах: для системы, используемой в жилом помещении и для системы, используемой в складских помещениях. В зависимости от особенностей объектов автоматизации, данное руководство может быть изменено.

5 ВВЕДЕНИЕ СИСТЕМЫ В ЭКСПЛУАТАЦИЮ

5.1 Введение в эксплуатацию системы, используемой для эксплуатации в жилищных помещениях

5.1.1 Введение в эксплуатацию технических средств

В первую очередь предусматривается установка и монтаж оборудования в помещении, подлежащем идентификации объектов. Сюда относится:

- создание и настройка беспроводной сети Wi-Fi (при необходимости);
- установка и настройка компьютера-моноблока TDC-5000PC;
- оснащение объектов, подлежащих идентификации, RFID-метками необходимого типа, в зависимости от особенностей объектов (материал, форма).

5.1.2 Введение в эксплуатацию программного обеспечения

Сюда относится:

- создание базы данных (внесение в систему объектов, подлежащих идентификации, оформление информации о владельце объекта);
- установка и настройка программ.

5.1.3 Обучение пользователей эксплуатации системы

Сюда входит обучение пользователей работе с программами, базой данных: добавлению новых объектов в базу данных, удалению имеющихся объектов из системы; внесению информации о пользователе: фамилия, имя, отчество, адрес и контактный телефон, по номеру которого будут приходить оповещения от системы.

5.1.4 Опытная эксплуатация

Опытная эксплуатация системы включает в себя проверку корректной работы всех компонентов системы:

- проверка целостности базы данных;
- проверка работы компьютера-моноблока и его соединения по беспроводной сети Wi-Fi;

- проверка работы встроенного RFID-считывателя.

При необходимости проводится отладка и корректировка системы.

Затем следует сдача и подписание актов приемки-сдачи работ. Также необходимо составить договор о последующем обслуживании системы в ходе повседневной эксплуатации.

5.2 Введение в эксплуатацию системы, используемой в складских помещениях

5.2.1 Введение в эксплуатацию технических средств

Система предусматривает интеграцию в существующую систему предприятия (склада), например такую как 1С. В противном случае база данных системы идентификации устанавливается на сервер.

Также необходимо создание и настройка беспроводной сети Wi-Fi. Затем установка и настройка считывателей системы и принтера RFID-меток.

Для опытной эксплуатации также потребуется оснащение небольшого количества товара RFID-метками.

5.2.2 Введение в эксплуатацию программного обеспечения

Сюда относится:

- создание базы данных (внесение в систему объектов, подлежащих идентификации);
- установка и настройка программ.

5.2.3 Обучение пользователей эксплуатации системы

Сюда входит обучение пользователей работе с программами, базой данных, в зависимости от занимаемой должности пользователей.

5.2.4 Опытная эксплуатация

Необходима демонстрация работы системы и проверка корректной работы и взаимодействия всех компонентов системы.

При возникновении сбоя работы системы производится отладка и корректировка системы. После чего, если не возникает претензий со стороны заказчика, следует сдача и подписание актов приемки-сдачи работ. Для последующего сопровождения системы в ходе повседневной эксплуатации составляется новый договор.

Вывод по разделу пять:

В данной главе было написано, как именно происходит ввод системы в эксплуатацию в двух вариантах: для системы, используемой в жилом помещении и для системы, используемой на складе. Ввод в эксплуатацию системы может быть изменен, в зависимости от особенностей объектов автоматизации.

6 ТЕХНИКО-ЭКОНОМИЧЕСКИЕ ПОКАЗАТЕЛИ СИСТЕМЫ

6.1 Технико-экономические показатели

6.1.1 Примерная калькуляция технических средств системы, используемой в жилом помещении

Примерная калькуляция технических средств системы, используемой в жилом помещении, приведена в таблице 6.1.

Таблица 6.1 - Примерная калькуляция технических средств системы, используемой в жилом помещении

Наименование	Цена (руб.)	Количество	Стоимость (руб.)
Активная метка CS3151TC	3 750,00	1	3 750,00
Пассивная метка на клеевой основе ALN-9740 'Squiggle'	12,5	20	250,00
Пассивная метка IL-07k.	576,00	2	1152,00
Компьютер-моноблок TDC-5000PC	126000,00	1	126000,00
ZYXEL Keenetic Lite III	1720,00	1	1720,00

Итого: 132 872 рубля. Стоимость системы зависит от особенностей объекта автоматизации.

6.1.2 Примерная калькуляция технических средств системы, используемой в складских помещениях

Примерная калькуляция технических средств системы, используемой в складских помещениях, приведена в таблице 6.2.

Стоимость системы зависит от особенностей склада и количества объектов, подлежащих идентификации.

Таблица 6.2 - Примерная калькуляция технических средств системы, используемой в складских помещениях

Наименование	Цена (руб.)	Количество	Стоимость (руб.)
RFID Считыватель FEIG LRU3000 UHF	205 640,00	1	205 640,00
Компьютер-моноблок TDC-5000PC	126000,00	1	126000,00
Считыватель мобильный FEIG PRHD102-USB HF/UHF	60 992,00	1	60 992,00
Промышленный RFID-принтер этикеток ZEBRA R110Xi4	215 419,00	1	215 419,00
Самоклеящаяся RFID-метка MIKRON M5454(2) M-3D UHF	30,00	2000	60000,00
Маршрутизатор Keenetic Giga III	5 080,00	1	5 080,00

Итого: 673 131 рубль.

Вывод по разделу шесть:

Стоимость системы (по итогам примерной калькуляции) в двух вариантах: для системы, используемой в жилом помещении и для системы, используемой на складе, является вполне приемлемой.

ЗАКЛЮЧЕНИЕ

В данной работе была разработана система контроля объектов в ограниченном объеме в двух вариантах: использование системы в жилых помещениях и в складских помещениях.

Система основана на вычислительной концепции IOT и реализована посредством технологии радиочастотной идентификации объектов.

В ходе работы разработана база данных и написан интерфейс приложения для взаимодействия пользователя, RFID-считывателя и базы данных. Компоненты системы (сервер и клиентская часть) работают корректно.

Внедрение данной системы в производственной сфере оптимизирует рабочее время и выявляет неэффективные звенья в цепи производственного процесса за счет отслеживания перемещений объектов, а также сокращает временные затраты на поиск нужного объекта.

При внедрении системы в жилом помещении сокращается время пользователя на поиск необходимого объекта, а также оптимизируются временные затраты за счет оповещения системой пользователя, при отсутствии объекта идентификации в сумке.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1 Карты и RFID технология, интернет-страница компании «Credit-card», http://credit-card.ru/articles/technology/rfid-cards.php.

2 Шарфельд Т. Системы RFID низкой стоимости, М.: «Мир»,2006.

3 RFID-технологии в производстве // Журнал «ИСУП», Горбунов А.О. - 2005. -№4. - стр. 8-9.

4 RFID технологии, интернет-страница компании «IQSKLAD» http://www.iqsklad.ru/rfid.

5 Области применения RFID считывателей, интернет-страница компании "ЦЕНТР КТ", http://shtrih-center.ru/state/primenenie_rfid_schityvatelei.html.

6 Идентификация потоков грузов RFID-метками, интернет-страница компании «Wireless technolgies» http://www.wirelesse.ru/articles/bluetooth/2011_02_34.php.

7 Бхуптани М., Морадпур Ш. RFID-технологии на службе вашего бизнеса - М.: «Альпина Паблишер», 2007.

8 Лахири С. RFID. Руководство по внедрению – М.: «Кудис-Пресс», 2007.

9 RFID-технология на складе, интернет-страница компании «Склад менеджмент», http://www.sklad-man.ru/page.php?id=220.

10 М. Федоров, Технология RFID. Опыт использования и перспективные направления, «Компоненты и технологии» №9, 2005 г.

11 RFID или штрих-код для автоматической идентификации, интернет-страница компании «СпектрПрестиж+» http://www.spe trprestig.ru/stati/ ontrolupravleniya-dostupom/rfid-ili-shtrih-kod.html.

12 RFID-технология, интернет-страница компании «РСТ-Инвент», http://www.rst-invent.ru/about/technology/.

13 Логистика и склад, интернет-страница компании «Релвест» http://www.rfidspb.ru/356/460/.

14 Из чего состоит система RFID, интернет-страница компании «ПОС-Магазин», http://posmagazin.ru/articles/18/120/.

15 RFID, интернет-страница «Традиция», www.traditio-ru.org/wiki/RFID.

16 Технология RFID, интернет-страница компании ООО «Энтерра Софт», http://www.enterra.ru/blog/rfid-implementation-guide-for-retail-with-technicaladdendum/.

17 Стандарт IEEE 802.15.4f-2012. RFID. – М: Издательство Витал Электроникс, 2011.

18 Применение RFID для инвентаризации основных средств, интернет-страница компании «РСТ-Инвент», http://www.rst-invent.ru/solv/ostmc/.

19 RFID метки: MIKRON M5454(2) M-3D UHF, интернет-страница компании «ISBC», http://www.isbc-rfid.ru/catalog/rfid-metki/uhf-865-965-mhz-epc-c1gen2-iso-18000-6c/mikron-m5454-2-m-3d-uhf/.

20 RFID метки: RFID-наклейка на пленке, интернет-страница компании «ISBC», http://www.isbc-rfid.ru/catalog/rfid-metki/uhf-865-965-mhz-epc-c1gen2-iso-18000-6c/rfid-naklejka-na-pljonke/.

21 RFID метки: Активная метка CS3151TC, интернет-страница компании «ISBC», http://www.isbc-rfid.ru/catalog/rfid-metki/rtls-microwave-active-2-44-ghz/aktivnaja-metka-cs3151tc/.

22 RFID-считыватель: FEIG PRHD102-USB HF/UHF, Интернет-страница компании «ISBC» http://www.isbc-rfid.ru/catalog/rfid-schityvateli/uhf-rfid-schityvateli/feig-prhd102-usb-hfuhf/.

23 RFID-считыватель: RFID Считыватель FEIG LRU3000 UHF, Интернет-страница компании «ISBC» http://www.isbc-rfid.ru/catalog/rfid-schityvateli/uhf-rfid-schityvateli/rfid-schityvatel-feig-lru3000-uhf/.

24 Компьютер-моноблок, интернет-страница компании «ИнтраСофт» http://www.intrasoft-spb.ru/catalog/computers/monitor-kbd-industrial/tdc-5000pc-kompyuter-monoblok-s-sensorny.php.

25 RFID-принтер: Zebra RZ400, интернет-страница компании «I SKLAD», http://www.iqsklad.ru/catalog/model/1889.html.

ПРИЛОЖЕНИЕ А

Листинг исходного кода:

YourBags:

program.cs:

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Threading.Tasks;
using System.Windows.Forms;

namespace YourBags
{
    static class Program
    {
        /// <summary>
        /// Главная точка входа для приложения.
        /// </summary>
        [STAThread]
        static void Main()
        {
            Application.EnableVisualStyles();
            Application.SetCompatibleTextRenderingDefault(false);
            Application.Run(new MainForm());
        }
    }
}
```

Form1.cs:

```
using System;
```

```
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;
using System.Net;
using System.IO;
using System.Web;
using System.Threading;

namespace YourBags
{
    public partial class MainForm : Form
    {
        public MainForm()
        {
            InitializeComponent();
            DCDataContext DC = new DCDataContext();
        }

        private void richTextBox1_TextChanged(object sender, EventArgs e)
        {
            DCDataContext DB1 = new DCDataContext();

            //Сохраним позицию курсора и выделение
            int saveSelPos = richTextBox1.SelectionStart;
```

```
int saveSelLen = richTextBox1.SelectionLength;

//Последовательный пойск всех элементов
int position = 0;
int idBags = 1;

var bag = from Your_Bags in DB1.Your_Bags
        select new { Your_Bags.Метка_сумки };

foreach (var Your_Bags in bag)
{
   string findbag = Your_Bags.Метка_сумки;
   //textBox1.Text = findbag;
   position = richTextBox1.Find(findbag, RichTextBoxFinds.WholeWord);
   if (position != -1)
      break;
   position = 0;
   idBags++;
}

string idbag = idBags.ToString();

string[] forget;
forget = new string[20];

for (int i = 0; i < 20; i++)
{
   forget[i] = "";
}
```

```
int K = 0;

var obj = from Your_Objects in DB1.Your_Objects
        where (Your_Objects.id_Сумки == idbag)
        select new { Your_Objects.Код_RFID_метки,
Your_Objects.Наименование_объекта };

foreach (var Your_Objects in obj)
{
   string findobj = Your_Objects.Код_RFID_метки;
   //textBox1.Text = findbag;
   position = richTextBox1.Find(findobj, RichTextBoxFinds.WholeWord);
   if (position == -1)
   {
      forget[K] = Your_Objects.Наименование_объекта;
      K++;
   }
   position = 0;
}
K = 0;
string send = "Вы забыли:";
string to = "";
string _from = "YourBag";
string apikey =
"5DM2JKL1WL1LSK95MET44BR3CB073L0RQ43N65TX61M3ZB7PPV96XZ158
LM1RN98";
string url;
```

```
            if (forget[0] != "")
            {
                var number = from Your_Bags in DB1.Your_Bags
                        join Bag_s_Master in DB1.Bag_s_Master on
Your_Bags.id_Хозяина_сумки equals Bag_s_Master.id_Хозяина_сумки
                        where (Your_Bags.id_Сумки == idbag)
                        select new { Bag_s_Master.Телефон };

                foreach (var Bag_s_Master in number)
                {
                    double numb = Bag_s_Master.Телефон;
                    to = numb.ToString();
                }

                send = send + forget[0] + " " + forget[1] + " " + forget[2] + " " +
forget[3] + " " + forget[4] + " "
                    + forget[5] + " " + forget[6] + " " + forget[7] + " " + forget[8] + " " +
forget[9] + " "
                    + forget[10] + " " + forget[11] + " " + forget[12] + " " + forget[13] +
" " + forget[14] + " "
                    + forget[15] + " " + forget[16] + " " + forget[17] + " " + forget[18] +
" " + forget[19];

                url = "http://smspilot.ru/api.php" +
                "?send=" + Uri.EscapeUriString(send) +
                "&to=" + to +
                "&from=" + _from +
                "&apikey=" + apikey; // +
                    //"&charset=windows-1251";
```

```
            HttpWebRequest myHttpWebRequest =
            (HttpWebRequest)HttpWebRequest.Create(url);

            // выполняем запрос
            HttpWebResponse myHttpWebResponse =
(HttpWebResponse)myHttpWebRequest.GetResponse();
        }

        //Востановим позицию курсора и выделение
        richTextBox1.SelectionStart = saveSelPos;
        richTextBox1.SelectionLength = saveSelLen;
        Thread.Sleep(1800);
        richTextBox1.Clear();
    }

    private void button1_Click(object sender, EventArgs e)
    {
        Form2 f2 = new Form2();
        f2.Show();
    }

    private void button2_Click(object sender, EventArgs e)
    {
        Close();
    }
  }
}
```

Form2.cs:

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;
using System.Data.SqlClient;

namespace YourBags
{
    public partial class Form2 : Form
    {
        SqlDataAdapter da;
        public Form2()
        {
            InitializeComponent();
            DCDataContext DC = new DCDataContext();
            comboBox1.DisplayMember = "Таблица";
            comboBox1.ValueMember = "Таблица";
        }



        private void button1_Click(object sender, EventArgs e)
        {
            DataSet dataset = new DataSet();
```

```
            SqlConnection connection = new SqlConnection();
            SqlCommand comand = new SqlCommand();
            connection.ConnectionString = (@"Data
Source=LENOVOG500\SQLEXPRESS;Integrated Security=SSPI;Persist Security
Info=False;Initial Catalog=Your_Bags");
            DCDataContext DB1 = new DCDataContext();
            var selected = comboBox1.SelectedItem;
            string select = selected.ToString();
            if ( select == "Bag's Color")
            {
               da = new SqlDataAdapter("SELECT * FROM Bag_s_Color",
connection);
               SqlCommandBuilder cb = new SqlCommandBuilder(da);
               DataTable dt = new DataTable();
               dataGridView1.ReadOnly = false;
               dataGridView1.AllowUserToAddRows = true;
               dataGridView1.AllowUserToDeleteRows = true;
               da.Fill(dt);
               dataGridView1.DataSource = dt;
            }
            if ( select == "Bag's Master")
            {
               da = new SqlDataAdapter("SELECT * FROM Bag_s_Master",
connection);
               SqlCommandBuilder cb = new SqlCommandBuilder(da);
               DataTable dt = new DataTable();
               dataGridView1.ReadOnly = false;
               dataGridView1.AllowUserToAddRows = true;
               dataGridView1.AllowUserToDeleteRows = true;
```

```
            da.Fill(dt);
            dataGridView1.DataSource = dt;
        }
        if ( select == "Bag's Type")
        {
            da = new SqlDataAdapter("SELECT * FROM Bag_s_Type",
connection);
            SqlCommandBuilder cb = new SqlCommandBuilder(da);
            DataTable dt = new DataTable();
            dataGridView1.ReadOnly = false;
            dataGridView1.AllowUserToAddRows = true;
            dataGridView1.AllowUserToDeleteRows = true;
            da.Fill(dt);
            dataGridView1.DataSource = dt;
        }
        if ( select == "Your Bags" )
        {
            da = new SqlDataAdapter("SELECT * FROM Your_Bags", connection);
            SqlCommandBuilder cb = new SqlCommandBuilder(da);
            DataTable dt = new DataTable();
            dataGridView1.ReadOnly = false;
            dataGridView1.AllowUserToAddRows = true;
            dataGridView1.AllowUserToDeleteRows = true;
            da.Fill(dt);
            dataGridView1.DataSource = dt;
        }
        if ( select == "Your Objects")
        {
```

```
            da = new SqlDataAdapter("SELECT * FROM Your_Objects",
connection);
            SqlCommandBuilder cb = new SqlCommandBuilder(da);
            DataTable dt = new DataTable();
            dataGridView1.ReadOnly = false;
            dataGridView1.AllowUserToAddRows = true;
            dataGridView1.AllowUserToDeleteRows = true;
            da.Fill(dt);
            dataGridView1.DataSource = dt;
        }
    }

    private void button2_Click(object sender, EventArgs e)
    {
        da.Update((DataTable)dataGridView1.DataSource);
    }

    private void button3_Click(object sender, EventArgs e)
    {
        DataSet dataset = new DataSet();
        SqlConnection connection = new SqlConnection();
        SqlCommand comand = new SqlCommand();
        connection.ConnectionString = (@"Data
Source=LENOVOG500\SQLEXPRESS;Integrated Security=SSPI;Persist Security
Info=False;Initial Catalog=Your_Bags");
        DCDataContext DB1 = new DCDataContext();
        var selected = comboBox1.SelectedItem;
        string select = selected.ToString();
        if (select == "Bag's Color")
```

```
            {
                da = new SqlDataAdapter("SELECT * FROM Bag_s_Color",
connection);
                SqlCommandBuilder cb = new SqlCommandBuilder(da);
                DataTable dt = new DataTable();
                dataGridView1.ReadOnly = false;
                dataGridView1.AllowUserToAddRows = true;
                dataGridView1.AllowUserToDeleteRows = true;
                da.Fill(dt);
                dataGridView1.DataSource = dt;
            }
            if (select == "Bag's Master")
            {
                da = new SqlDataAdapter("SELECT * FROM Bag_s_Master",
connection);
                SqlCommandBuilder cb = new SqlCommandBuilder(da);
                DataTable dt = new DataTable();
                dataGridView1.ReadOnly = false;
                dataGridView1.AllowUserToAddRows = true;
                dataGridView1.AllowUserToDeleteRows = true;
                da.Fill(dt);
                dataGridView1.DataSource = dt;
            }
            if (select == "Bag's Type")
            {
                da = new SqlDataAdapter("SELECT * FROM Bag_s_Type",
connection);
                SqlCommandBuilder cb = new SqlCommandBuilder(da);
                DataTable dt = new DataTable();
```

```
            dataGridView1.ReadOnly = false;
            dataGridView1.AllowUserToAddRows = true;
            dataGridView1.AllowUserToDeleteRows = true;
            da.Fill(dt);
            dataGridView1.DataSource = dt;
        }
        if (select == "Your Bags")
        {
            da = new SqlDataAdapter("SELECT * FROM Your_Bags", connection);
            SqlCommandBuilder cb = new SqlCommandBuilder(da);
            DataTable dt = new DataTable();
            dataGridView1.ReadOnly = false;
            dataGridView1.AllowUserToAddRows = true;
            dataGridView1.AllowUserToDeleteRows = true;
            da.Fill(dt);
            dataGridView1.DataSource = dt;
        }
        if (select == "Your Objects")
        {
            da = new SqlDataAdapter("SELECT * FROM Your_Objects",
connection);
            SqlCommandBuilder cb = new SqlCommandBuilder(da);
            DataTable dt = new DataTable();
            dataGridView1.ReadOnly = false;
            dataGridView1.AllowUserToAddRows = true;
            dataGridView1.AllowUserToDeleteRows = true;
            da.Fill(dt);
            dataGridView1.DataSource = dt;
        }
```

```
        }

        private void button4_Click(object sender, EventArgs e)
        {
            Close();
        }
    }
}
```

ПРИЛОЖЕНИЕ Б

Logistic:

Program.cs:

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Threading.Tasks;
using System.Windows.Forms;

namespace Logistic
{
    static class Program
    {
        /// <summary>
        /// Главная точка входа для приложения.
        /// </summary>
        [STAThread]
        static void Main()
        {
            Application.EnableVisualStyles();
            Application.SetCompatibleTextRenderingDefault(false);
            Application.Run(new Form1());
        }
    }
}
```

Form1.cs:

```
using System;
using System.Collections.Generic;
```

```
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;
using System.Data.SqlClient;

namespace Logistic
{
    public partial class Form1 : Form
    {
        SqlDataAdapter da;
        public Form1()
        {
            InitializeComponent();
        }

        private void Form1_Load(object sender, EventArgs e)
        {
            DataSet dataset = new DataSet();
            SqlConnection connection = new SqlConnection();
            SqlCommand comand = new SqlCommand();
            connection.ConnectionString = (@"Data
Source=LENOVOG500\SQLEXPRESS;Integrated Security=SSPI;Persist Security
Info=False;Initial Catalog=Storage");
            da = new SqlDataAdapter("SELECT * FROM Товар", connection);
            SqlCommandBuilder cb = new SqlCommandBuilder(da);
```

```
            DataTable dt = new DataTable();
            dataGridView1.ReadOnly = false;
            dataGridView1.AllowUserToAddRows = true;
            dataGridView1.AllowUserToDeleteRows = true;
            da.Fill(dt);
            dataGridView1.DataSource = dt;
        }

        private void button1_Click(object sender, EventArgs e)
        {
            da.Update((DataTable)dataGridView1.DataSource);
        }

        private void button2_Click(object sender, EventArgs e)
        {
            DataSet dataset = new DataSet();
            SqlConnection connection = new SqlConnection();
            SqlCommand comand = new SqlCommand();
            connection.ConnectionString = (@"Data
Source=LENOVOG500\SQLEXPRESS;Integrated Security=SSPI;Persist Security
Info=False;Initial Catalog=Storage");
            da = new SqlDataAdapter("SELECT * FROM Товар", connection);
            SqlCommandBuilder cb = new SqlCommandBuilder(da);
            DataTable dt = new DataTable();
            dataGridView1.ReadOnly = false;
            dataGridView1.AllowUserToAddRows = true;
            dataGridView1.AllowUserToDeleteRows = true;
            da.Fill(dt);
            dataGridView1.DataSource = dt;
```

```
        }

        private void button3_Click(object sender, EventArgs e)
        {
            Close();
        }
    }
}
```

ScanerFrame:

Program.cs:

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Threading.Tasks;
using System.Windows.Forms;

namespace ScanerFrame
{
    static class Program
    {
        /// <summary>
        /// Главная точка входа для приложения.
        /// </summary>
        [STAThread]
        static void Main()
        {
            Application.EnableVisualStyles();
            Application.SetCompatibleTextRenderingDefault(false);
```

```
            Application.Run(new Form1());
        }
    }
}

Form1.cs:
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;
using System.Data.SqlClient;
using System.Threading;

namespace ScanerFrame
{
    public partial class Form1 : Form
    {
        public Form1()
        {
            InitializeComponent();
        }

        private void button1_Click(object sender, EventArgs e)
        {
```

```
    Close();
}

private void richTextBox1_TextChanged(object sender, EventArgs e)
{
    DCDataContext DB1 = new DCDataContext();

    //Сохраним позицию курсора и выделение
    int saveSelPos = richTextBox1.SelectionStart;
    int saveSelLen = richTextBox1.SelectionLength;

    //Последовательный поиск всех элементов
    int position = 0;
    int idObj = 1;

    var obj = from Товар in DB1.Товар
              select new { Товар.RFID_код_товара };

    foreach (var Товар in obj)
    {
        string findobj = Товар.RFID_код_товара;
        //textBox1.Text = findbag;
        position = richTextBox1.Find(findobj, RichTextBoxFinds.WholeWord);
        if (position != -1)
            break;
        position = 0;
        idObj++;
    }
```

```
            string idobj = idObj.ToString();
            string Checking = "";

            var check = from Товар in DB1.Товар
                      where ( Товар.id_товара == idobj )
                    select new { Товар.На_складе };

            foreach (var Товар in check)
            {
              var findcheck = Товар.На_складе;
              Checking = findcheck.ToString();
            }

            //textBox1.Text = Checking;

            SqlConnection connection = new SqlConnection();
            connection.ConnectionString = (@"Data
Source=LENOVOG500\SQLEXPRESS;Integrated Security=SSPI;Persist Security
Info=False;Initial Catalog=Storage");
            connection.Open();
            string sqlfalse = string.Format("Update Товар Set На_складе = '{0}'
Where id_товара = '{1}'", 0, idObj);
            string sqltrue = string.Format("Update Товар Set На_складе = '{0}' Where
id_товара = '{1}'", 1, idObj);

            if (Checking == "True")
            {
               using (SqlCommand cmd = new SqlCommand(sqlfalse, connection))
               {
```

```
                cmd.ExecuteNonQuery();
            }
        }
        if (Checking == "False")
        {
            using (SqlCommand cmd = new SqlCommand(sqltrue, connection))
            {
                cmd.ExecuteNonQuery();
            }
        }
        position = 0;
        richTextBox1.SelectionStart = saveSelPos;
        richTextBox1.SelectionLength = saveSelLen;
        connection.Close();
        Thread.Sleep(1800);
        richTextBox1.Clear();

    }
  }
}
```

Loader:

Program.cs:

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Threading.Tasks;
using System.Windows.Forms;
```

```
namespace loader
{
    static class Program
    {
        /// <summary>
        /// Главная точка входа для приложения.
        /// </summary>
        [STAThread]
        static void Main()
        {
            Application.EnableVisualStyles();
            Application.SetCompatibleTextRenderingDefault(false);
            Application.Run(new MainForm());
        }
    }
}
```

MainForm:

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;

namespace loader
```

```
{
    public partial class MainForm : Form
    {
        public MainForm()
        {
            InitializeComponent();
        }

        private void button1_Click(object sender, EventArgs e)
        {
            Form1 f1 = new Form1();
            f1.Show();
        }

        private void button2_Click(object sender, EventArgs e)
        {
            Form2 f2 = new Form2();
            f2.Show();
        }

        private void button3_Click(object sender, EventArgs e)
        {
            Close();
        }
    }
}
```

Form1.cs:

```
using System;
```

```
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;
using System.Data.SqlClient;

namespace loader
{
    public partial class Form1 : Form
    {
        SqlDataAdapter da;
        public Form1()
        {
            InitializeComponent();
        }

        private void Form1_Load(object sender, EventArgs e)
        {
            DataSet dataset = new DataSet();
            SqlConnection connection = new SqlConnection();
            SqlCommand comand = new SqlCommand();
            connection.ConnectionString = (@"Data
Source=LENOVOG500\SQLEXPRESS;Integrated Security=SSPI;Persist Security
Info=False;Initial Catalog=Storage");
```

```
            da = new SqlDataAdapter("SELECT Занято, id_места FROM
Место_на_полке WHERE Занято = 0", connection);
            SqlCommandBuilder cb = new SqlCommandBuilder(da);
            DataTable dt = new DataTable();
            dataGridView2.ReadOnly = false;
            dataGridView2.AllowUserToAddRows = true;
            dataGridView2.AllowUserToDeleteRows = true;
            da.Fill(dt);
            dataGridView2.DataSource = dt;
            DCDataContext DB1 = new DCDataContext();
            bool check = false;
            var place = from Место_на_полке in DB1.Место_на_полке
                    join Полка in DB1.Полка on Место_на_полке.id_полки equals
Полка.id_полки
                    join Стеллаж in DB1.Стеллаж on Полка.id_стеллажа equals
Стеллаж.id_стеллажа
                    join Ряд in DB1.Ряд on Стеллаж.id_ряда equals Ряд.id_ряда
                    where (Место_на_полке.Занято == check)
                    select new { Ряд.Номер_ряда, Стеллаж.Номер_стеллажа,
Полка.Номер_полки, Место_на_полке.Номер_места };
            dataGridView1.DataSource = place;
        }

        private void button1_Click(object sender, EventArgs e)
        {
            da.Update((DataTable)dataGridView2.DataSource);
            DCDataContext DB1 = new DCDataContext();
            DataSet dataset = new DataSet();
            SqlConnection connection = new SqlConnection();
```

```
            SqlCommand comand = new SqlCommand();
            connection.ConnectionString = (@"Data
Source=LENOVOG500\SQLEXPRESS;Integrated Security=SSPI;Persist Security
Info=False;Initial Catalog=Storage");

            int saveSelPos = richTextBox1.SelectionStart;
            int saveSelLen = richTextBox1.SelectionLength;

            //Последовательный пойск всех элементов
            int position = 0;
            string findobj = "";
            string idObj = "";
            string findplace = "";
            string idPlace = "";

            var obj = from Товар in DB1.Товар
                    select new { Товар.RFID_код_товара, Товар.id_товара };

            foreach (var Товар in obj)
            {
               findobj = Товар.RFID_код_товара;
               idObj = Товар.id_товара;
               //textBox1.Text = findbag;
               position = richTextBox1.Find(findobj, RichTextBoxFinds.WholeWord);
               if (position != -1)
                  break;
               position = 0;
            }
            position = 0;
```

```
            var placement = from Место_на_полке in DB1.Место_на_полке
                    select new { Место_на_полке.RFID_код_места,
Место_на_полке.id_места };

            foreach (var Место_на_полке in placement)
            {
                findplace = Место_на_полке.RFID_код_места;
                idPlace = Место_на_полке.id_места;
                //textBox1.Text = findbag;
                position = richTextBox1.Find(findplace,
RichTextBoxFinds.WholeWord);
                if (position != -1)
                    break;
                position = 0;
            }
            position = 0;


            var s = from Товар_Место in DB1.Товар_Место
                    select new { Товар_Место.id_записи };
            int id = s.Count();
            id++;

            connection.Open();
            string sql = string.Format("Insert Into Товар_Место" +
                "(id_записи, id_товара, RFID_код_товара, id_места,
RFID_код_места) Values(@id_записи, @id_товара, @RFID_код_товара,
@id_места, @RFID_код_места)");
```

```
using (SqlCommand cmd = new SqlCommand(sql, connection))
{
   cmd.Parameters.AddWithValue("@id_записи", id);
   cmd.Parameters.AddWithValue("@id_товара", idObj);
   cmd.Parameters.AddWithValue("@RFID_код_товара", findobj);
   cmd.Parameters.AddWithValue("@id_места", idPlace);
   cmd.Parameters.AddWithValue("@RFID_код_места", findplace);

   cmd.ExecuteNonQuery();
}
idObj = "";
idPlace = "";
connection.Close();

   da = new SqlDataAdapter("SELECT Занято FROM Место_на_полке
WHERE Занято = 0", connection);
SqlCommandBuilder cb = new SqlCommandBuilder(da);
DataTable dt = new DataTable();
dataGridView2.ReadOnly = false;
dataGridView2.AllowUserToAddRows = true;
dataGridView2.AllowUserToDeleteRows = true;
da.Fill(dt);
dataGridView2.DataSource = dt;

bool check = false;
var place = from Место_на_полке in DB1.Место_на_полке
        join Полка in DB1.Полка on Место_на_полке.id_полки equals
Полка.id_полки
```

```
                    join Стеллаж in DB1.Стеллаж on Полка.id_стеллажа equals
Стеллаж.id_стеллажа
                    join Ряд in DB1.Ряд on Стеллаж.id_ряда equals Ряд.id_ряда
                    where (Место_на_полке.Занято == check)
                    select new { Ряд.Номер_ряда, Стеллаж.Номер_стеллажа,
Полка.Номер_полки, Место_на_полке.Номер_места };
            dataGridView1.DataSource = place;
        }

        private void button2_Click(object sender, EventArgs e)
        {
            richTextBox1.Clear();
            DataSet dataset = new DataSet();
            SqlConnection connection = new SqlConnection();
            SqlCommand comand = new SqlCommand();
            connection.ConnectionString = (@"Data
Source=LENOVOG500\SQLEXPRESS;Integrated Security=SSPI;Persist Security
Info=False;Initial Catalog=Storage");
            da = new SqlDataAdapter("SELECT Занято FROM Место_на_полке
WHERE Занято = 0", connection);
            SqlCommandBuilder cb = new SqlCommandBuilder(da);
            DataTable dt = new DataTable();
            dataGridView2.ReadOnly = false;
            dataGridView2.AllowUserToAddRows = true;
            dataGridView2.AllowUserToDeleteRows = true;
            da.Fill(dt);
            dataGridView2.DataSource = dt;
            DCDataContext DB1 = new DCDataContext();
            bool check = false;
```

```
            var place = from Место_на_полке in DB1.Место_на_полке
                    join Полка in DB1.Полка on Место_на_полке.id_полки equals
Полка.id_полки
                    join Стеллаж in DB1.Стеллаж on Полка.id_стеллажа equals
Стеллаж.id_стеллажа
                    join Ряд in DB1.Ряд on Стеллаж.id_ряда equals Ряд.id_ряда
                    where (Место_на_полке.Занято == check)
                    select new { Ряд.Номер_ряда, Стеллаж.Номер_стеллажа,
Полка.Номер_полки, Место_на_полке.Номер_места };
            dataGridView1.DataSource = place;
        }

        private void button3_Click(object sender, EventArgs e)
        {
            Close();
        }
    }
}
```

Form2.cs:

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;
```

```
using System.Data.SqlClient;

namespace loader
{
    public partial class Form2 : Form
    {
        public Form2()
        {
            InitializeComponent();
            DCDataContext DC = new DCDataContext();
            comboBox1.DisplayMember = "id_товара";
            comboBox1.ValueMember = "id_товара";
            comboBox1.DataSource = DC.Товар.ToList();
        }

        private void button1_Click(object sender, EventArgs e)
        {
            DCDataContext DB1 = new DCDataContext();
            var selected = comboBox1.SelectedItem;
            string id = ((Товар)selected).id_товара;
            var s = from Товар in DB1.Товар
                    join Товар_Место in DB1.Товар_Место on Товар.id_товара equals
Товар_Место.id_товара
                    join Место_на_полке in DB1.Место_на_полке on
Товар_Место.id_места equals Место_на_полке.id_места
                    join Полка in DB1.Полка on Место_на_полке.id_полки equals
Полка.id_полки
                    join Стеллаж in DB1.Стеллаж on Полка.id_стеллажа equals
Стеллаж.id_стеллажа
```

```
                join Ряд in DB1.Ряд on Стеллаж.id_ряда equals Ряд.id_ряда
                where (Товар.id_товара == id)
                select new
                {
                    Ряд.Номер_ряда,
                    Стеллаж.Номер_стеллажа,
                    Полка.Номер_полки,
                    Место_на_полке.Номер_места,
                    Товар_Место.RFID_код_места,
                    Товар.Наименование_товара,
                    Товар_Место.RFID_код_товара
                };
            dataGridView1.DataSource = s;
        }

        private void button2_Click(object sender, EventArgs e)
        {
            DCDataContext DB1 = new DCDataContext();
            DataSet dataset = new DataSet();
            SqlConnection connection = new SqlConnection();
            SqlCommand comand = new SqlCommand();
            connection.ConnectionString = (@"Data
Source=LENOVOG500\SQLEXPRESS;Integrated Security=SSPI;Persist Security
Info=False;Initial Catalog=Storage");

            int saveSelPos = richTextBox1.SelectionStart;
            int saveSelLen = richTextBox1.SelectionLength;

            //Последовательный пойск всех элементов
```

```
            int position = 0;
            string findobj = "";
            string idObj = "";
            string findplace = "";
            string idPlace = "";

            var obj = from Товар_Место in DB1.Товар_Место
                    select new { Товар_Место.RFID_код_товара,
Товар_Место.id_записи };

            foreach (var Товар_Место in obj)
            {
               findobj = Товар_Место.RFID_код_товара;
               idObj = Товар_Место.id_записи;
               //textBox1.Text = findbag;
               position = richTextBox1.Find(findobj, RichTextBoxFinds.WholeWord);
               if (position != -1)
                  break;
               position = 0;
            }
            position = 0;

            var placement = from Место_на_полке in DB1.Место_на_полке
                        select new { Место_на_полке.RFID_код_места,
Место_на_полке.id_места };

            foreach (var Место_на_полке in placement)
            {
               findplace = Место_на_полке.RFID_код_места;
```

```
                idPlace = Место_на_полке.id_места;
                //textBox1.Text = findbag;
                position = richTextBox1.Find(findplace,
RichTextBoxFinds.WholeWord);
                if (position != -1)
                    break;
                position = 0;
            }
            position = 0;

            connection.Open();
            string sql = string.Format("Delete from Товар_Место where id_записи =
'{0}'", idObj);
            using (SqlCommand cmd = new SqlCommand(sql, connection))
            {
                cmd.ExecuteNonQuery();
            }

            string sqlfalse = string.Format("Update Место_на_полке Set Занято =
'{0}' Where id_места = '{1}'", 0, idPlace);
            using (SqlCommand cmd = new SqlCommand(sqlfalse, connection))
            {
                cmd.ExecuteNonQuery();
            }
            connection.Close();
        }

        private void button3_Click(object sender, EventArgs e)
        {
```

```
            Close();
        }
    }
}
```