|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Приднестровский государственный университет им. Т.Г. Шевченко | | | | | | | | | |
| Инженерно-технический институт | | | | | | | | | |
| Инженерно-технический факультет | | | | | | | | | |
| Кафедра информационных технологий и автоматизированного управления  производственными процессами | | | | | | | | | |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  | Допустить к защите | | | |
|  |  |  |  |  |  | зав. кафедрой ИТиАУПП, | | | |
|  |  |  |  |  |  | к.т.н., доцент | | | |
|  |  |  |  |  |  |  |  | Ю.А. Столяренко | |
|  |  |  |  |  |  | « \_\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_\_ 2017 г. | | | |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
| **ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА** | | | | | | | | | |
|  | | | | | | | | | |
| **тема: «РАЗРАБОТКА ИНФОРМАЦИОННОЙ**  **СИСТЕМЫ С ПРИМЕНЕНИЕМ ТЕХНОЛОГИИ RFID»** | | | | | | | | | |
|  | | | | | | | | | |
| БАКАЛАВРСКАЯ РАБОТА | | | | | | | | | |
|  | | | | | | | | | |
|  | | | | | | | | | |
| Направление 09.03.02 «Информационные системы и технологии» | | | | | | | | | |
| Профиль: «Безопасность информационных систем» | | | | | | | | | |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  | Студент | |  |  |  |  | Талпа | | |
|  | группы ИТ13ДР62ИС1 | | |  |  |  | Роман Витальевич | | |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  | Научный руководитель, | | |  |  |  | Столяренко | | |
|  | к.т.н., доцент | | |  |  |  | Юлия Александровна | | |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
| Тирасполь, 2017 | | | | | | | | | |

Приднестровский государственный университет им. Т.Г. Шевченко

Инженерно-технический институт

Инженерно-технический факультет

Кафедра информационных технологий и автоматизированного управления

производственными процессами

Утверждаю

зав. кафедрой ИТиАУПП,

к.т.н., доцент

Ю.А. Столяренко

**«\_\_\_\_» \_\_\_\_\_\_\_\_\_\_\_** 2017 г.

**ЗАДАНИЕ**

**НА ВЫПУСКНУЮ КВАЛИФИКАЦИОННУЮ РАБОТУ БАКАЛАВРА**

**Студенту** *Талпа Роману Витальевичу*

**Тема ВКРБ:** «*Разработка информационной системы с применением технологии RFID»*

утверждена приказом по университету № \_\_\_\_от «\_\_\_» \_\_\_\_2017 г.

Срок сдачи расчетно-пояснительной записки на кафедру «\_\_\_» \_\_\_\_ 2017г.

**Исходные данные к работе:** *структурная схема информационной базы данных, конфигурация программно-аппаратного обеспечения.*

**Перечень подлежащих разработке вопросов:** *разработать информационную систему реализующую хранение и обработку данных с возможностью проверки документа бесконтактной технологией.*

**Перечень дополнительных вопросов**: *произвести анализ поставленной задачи, найти оптимальный вариант построение программно-аппаратного продукта, рассмотрения вопросов охраны труда.*

Дата выдачи задания «\_\_\_\_» \_\_\_\_\_\_\_\_ 2017 г.

Научный руководитель, к.т.н., доцент \_\_\_\_\_\_\_\_/Столяренко Ю.А./

Задание принял к исполнению\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_/Талпа Р.В./

**АННОТАЦИЯ**

Целью дипломной работы была разработка информационной системы и программно-аппаратного комплекса для работы с информационной системой.

Информационная система должна удовлетворять ряду технических требований, таких как процесс хранения, заполнения и изменения данных.

Реализовать возможность бесконтактной проверки документа, а также облегчить обработку информации, реализовать возможность хранения данных на сервере.

**ABSTRACT**

The aim of the thesis the development of an information system and hardware-software complex for working with the information system

The information system must meet certain technical requirements, such as the process of storing, filling and changing data.

Implement the contactless inspection document, and to facilitate information processing, to realize the possibility of storing data on the server.

**ОГЛАВЛЕНИЕ**

|  |  |  |
| --- | --- | --- |
| ВВЕДЕНИЕ | | 5 |
| 1 ИССЛЕДОВАНИЕ И АНАЛИЗ ПРЕДМЕТНОЙ ОБЛАСТИ | | 6 |
|  | 1.1 Обоснование актуальности исследуемой задачи | 6 |
|  | 1.2 Современное состояние исследуемой задачи | 7 |
|  | 1.3 Обзор методов решения подобных задач | 7 |
|  | 1.4 Постановка задачи, системные требования, требования к входным  данным и выходным формам  1.5 Выводы | 11  12 |
| 2 РАЗРАБОТКА МЕТОДОВ РЕШЕНИЯ ПОСТАВЛЕННОЙ ЗАДАЧИ | | 13 |
|  | 2.1 Выбор методов и средств для реализации, его обоснование | 13 |
|  | 2.2 Описание логической структуры решения задачи | 20 |
|  | 2.3 Функциональная схема решения задачи  2.4 Выводы | 26  28 |
| 3 РЕАЛИЗАЦИЯ И ТЕСТИРОВАНИЕ ВЫБРАННЫХ МЕТОДОВ  РЕШЕНИЯ ЗАДАЧИ | | 29 |
|  | 3.1 Описание реализации | 29 |
|  | 3.2 Описание пользовательского интерфейса | 34 |
|  | 3.3 Методы и средства защиты метода решения задачи | 36 |
|  | 3.4 Тестирование и оценка надежности программного продукта | 37 |
|  | 3.5 Расчет себестоимости от внедрения результатов выпускной  квалификационной работы бакалавра | 39 |
|  | 3.6 Охрана труда  3.7 Выводы | 46  50 |
| ЗАКЛЮЧЕНИЕ | | 51 |
| ПЕРЕЧЕНЬ УСЛОВНЫХ ОБОЗНАЧЕНИЙ, СИМВОЛОВ, ЕДИНИЦ  И ТЕРМИНОВ | | 52 |
| СПИСОК ЛИТЕРАТУРЫ | | 53 |
| ПРИЛОЖЕНИЕ А – Руководство пользователя | | 54 |

**ВВЕДЕНИЕ**

Широкое распространение компьютерной техники позволяет автоматизировать обработку данных практически в любой сфере человеческой деятельности.

Одним из наиболее распространенных классов систем обработки данных являются информационные системы. Они используют ресурсы нескольких категорий средства вычислительной техники, системное и прикладное программное обеспечение, информационные, лингвистические и человеческие ресурсы. Их основной задачей является удовлетворение конкретных информационных потребностей в рамках конкретной предметной области.

Целью работы является разработка информационной системы, реализация удобства и простоты использования программно-аппаратного продукта.

Информационные системы обеспечивают сбор, хранение, обработку, поиск, выдачу информации, необходимой в процессе принятия решений задач из любой области. Они помогают анализировать проблемы и создавать новые продукты.

Так же применение технологии *RFID* (*Radio Frequency Identification* – радиочастотная идентификация) – это технология нового поколения, основанная на использовании радиочастотного электромагнитного излучения. *RFID* применяется для идентификации и учета объектов.

Исходя из данных минусов, такую проблему можно решить при помощи, бесконтактных технологий и информационной базы данных. Документ снабжен чипом с некой информацией которую можно считать при помощи радиочастотного электромагнитного излучения и в дальнейшем считывания. Это позволяет сэкономить время и исключить ошибки.

**1 ИССЛЕДОВАНИЕ И АНАЛИЗ ПРЕДМЕТНОЙ ОБЛАСТИ**

**1.1 Обоснование актуальности исследуемой задачи**

Рассматривается проблема разработки информационной системы с возможностью бесконтактной проверки документов. Данная информационная система предназначена для хранения, поиска и обработки данных о документах содержащие электронную информацию.

Информационные базы данных успешно используются в государственных деятельностях, учебных заведениях, бизнесах и т.д. В роли государственных деятельностей базы данных используется на таможнях, паспортном учете, министерстве внутренних дел.

Так же информационная база данных может взаимодействовать с бесконтактными технологиями. Технология бесконтактной идентификации широко применяется во всем мире. Для проезда в метрополитене требуется пропуск, с помощью которого можно пройти через турникет. Для оплаты какого-либо товара, применяется технология бесконтактной оплаты банковской картой, всего лишь требуется преподнести карту к терминалу оплаты. На складах огромный поток товара с которым можно управиться намного быстрее применяя такую технологию. Это позволяет ускорить работу и понизить различные ошибки человеческого фактора.

В настоящее время все чаще и чаще поступают жалобы об угоне транспортного средства. На посту попадаются водители с поддельными документами, поддельными государственными номерными знаками, поддельной доверенностью или угнанные транспортные средства.

Информационная система, а также технологии, которые будут применяться, рассчитаны на такие нарушения. Так как устройство, реализованное с применением технологии *RFID*, позволит в реальном времени проверить документ и найти информацию о документе в базе данных.

**1.2 Современное состояние исследуемой задачи**

Существует большое число различных информационных систем, предназначенных для решения структурированных задач в различных сферах. Информационная система включают в себя базу данных, в которой хранится вся информация и систему управления базой данных для взаимодействия с ней, а также средства поиска и сбора информации. Система позволяет упорядочить хранимые данные и осуществлять быстрый доступ к информации.

Помимо информационной базы данных также требуется применить радиочастотную идентификацию. Попытки решить задачу автоматической идентификации, например, транспорта предпринимаются уже несколько десятилетий. Для этого предлагаются различные технологии: штриховое кодирование, видеосистемы, способные распознавать информацию номерного знака, и многие другие. А в последние годы все больший интерес вызывают системы идентификации, построенные на базе технологии *RFID.*

**1.3 Обзор методов решения подобных задач**

Поскольку решаемая задача не является уникальной в своем роде, разумно рассмотреть различные варианты ее решения и определить оптимальный путь создания и развития новой системы. Опираясь на уже имеющиеся методы решения схожего ряда задач, удобно обозначить функциональные особенности тех или иных систем создания расписания, для того чтобы включить их в создаваемый программно-аппаратный продукт.

Так же в настоящее время известен ряд технологий бесконтактной идентификации, среди них:

* технологии штрихового кодирования;
* технологии радиочастотной идентификации;
* биометрические технологии;
* технологии машинного зрения;
* технологии речевого ввода данных и ряд других.

Исходя из вышеперечисленных технологий было решено использовать радиочастотную идентификацию. Метки также бывают активные и пассивные. Пассивные *RFID*-метки не имеют встроенного источника энергии. Электрический ток, индуцированный в антенне электромагнитным сигналом от считывателя, обеспечивает достаточную мощность для функционирования кремниевого чипа, размещённого в метке, и передачи ответного сигнала. Принцип считывания метки изображено на рисунке 1.1.

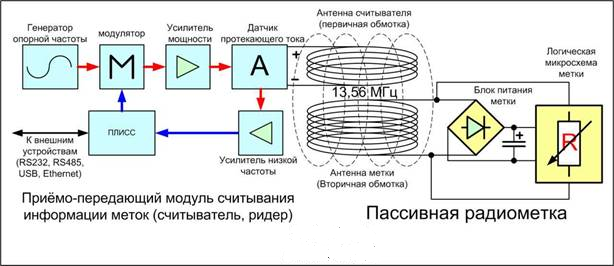


Рисунок 1.1 – Считывание пассивной метки

Рассмотрим наиболее известные принципы работы бесконтактной карты в разных сферах применения.

«Социальная карта москвича» – это многофункциональные бесконтактные чиповые карты, которые можно использовать как электронное удостоверение льготника, транспортную и банковскую платежную карту. Социальные карты москвича разработаны специально для пенсионеров и других категорий граждан, состоящих на учете в органах социальной защиты столицы, и направлены на повышение эффективности социальной поддержки населения Москвы.

Возможности социальной карты москвича:

* социальные карты москвича обеспечивают удобство реализации прав на льготный проезд в общественном транспорте города Москвы (автобус, троллейбус, трамвай, метрополитен) и Московской области, пригородном железнодорожном сообщении;
* социальные карты москвича позволяют обеспечить удобство и безопасность получения пенсий, стипендий, субсидий, пособий, городских социальных выплат и собственных средств;
* с помощью социальных карт москвича можно оплачивать различные услуги, в том числе жилищно-коммунальные, налоги, рассчитываться за товары в магазинах;
* более 3000 торгово-сервисных предприятий участвуют в дисконтной программе по предоставлению скидок лицам различных льготных категорий при оплате товаров и услуг;
* некоторые таксофоны столицы принимают социальные карты москвича для оплаты разговоров. Звонок можно осуществить при наличии денежных средств на счете социальной карты москвича;
* в ближайшее время владельцы социальных карт москвича получат возможность использовать их для оплаты услуг социального такси;
* предусмотрено использование Социальных карт москвича наравне с полисом обязательного медицинского страхования в городских лечебно-профилактических учреждениях.

Внешний вид социальной карты москвича изображено на рисунке 1.2.

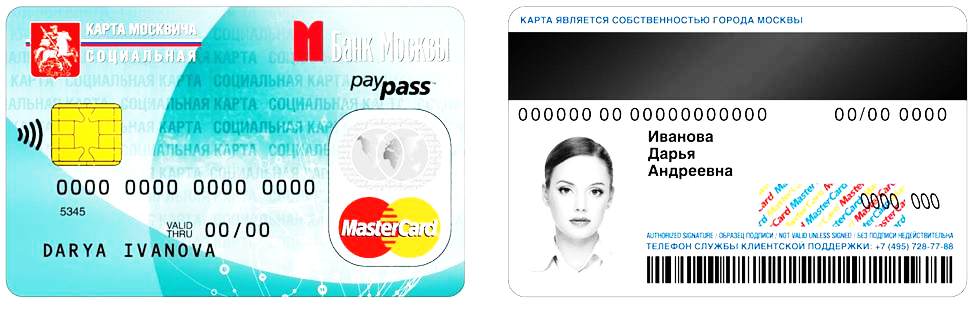
  
2

Рисунок 1.2 – Социальная карта москвича

Следующий вид технологии «электронная проходная» – это инновационное решение для организации проходов людей в офисных центрах, учреждениях, школах, университетах, на заводах и т.д.

Преимущества электронной проходной для руководителей организаций:

* формирование ответственного отношения персонала к режиму трудового дня: получение информации об опоздавших, рано ушедших, а также задержавшихся после окончания рабочего дня (это позволит выплачивать зарплату согласно реальному времени, проведенному на рабочем месте);
* получение отчета по рабочему времени для каждого работника за выбранный период;
* исключение прохода на предприятие сотрудников в их нерабочее время;
* оснащения сигнализацией от несанкционированного прохода.

Электронный пропуск и его применение:

* Доступ сотрудников по электронной пластиковой карте пропуску. Работники на проходной предъявляют электронные пропуска считывателям, размещенным на стойке турникета. При произведении этих действий, высвечивается наглядная пиктографическая индикация статуса турникета.
* Доступ посетителей. Системы и аналоги электронной проходной позволяют выдавать посетителям временные электронные пластиковые карты пропуски бесконтактного типа, имеющие статус «Посетитель». При входе пластик прикладывается к считывателю в турникете, выход производится после опускания карты в подсвеченное отверстие на крышке картоприемника. Пока посетитель не отдаст карточку, турникет не будет разблокирован.
* Ограничение доступа во внутренние помещения. У небольших компаний часто возникает потребность ограничить проход во внутренние помещения: кабинеты руководства, серверную, бухгалтерию, склад. Это особенно актуально, когда надо разделить помещения для посетителей и комнаты со служебным доступом. Эти системы намного надежней и удобней дверных замков. Пользование ключами вызывает у сотрудников неудобства, поэтому многие из них оставляют двери открытыми, отлучаясь ненадолго.

Принцип работы и взаимодействия пропускного билета в учебном заведении представлен на рисунке 1.3.

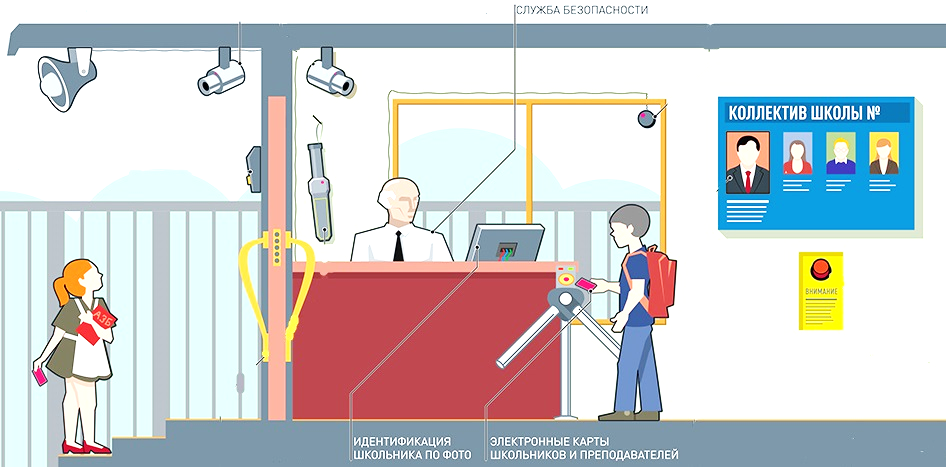


Рисунок 1.3 – Пропуск школьника через электронную проходную

Электронные пропуска в обложке и с фирменным логотипом будут выглядеть солидно и заменят неудобную связку ключей.

**1.4 Постановка задачи, системные требования, требования к входным   
данным и выходным формам  
1**

В результате изучения перечня существующих средств, реализующих механизм бесконтактного считывания, был выделен список функций и особенностей, который следует включить в разрабатываемый продукт, чтобы повысить уровень качества и наделить программу достаточным набором функций, в полной мере обеспечивающих ее всем необходимым для качественной проверки.

Исходя из описания поставленной задачи требуется выполнить следующий список задач:

* разработка программно-аппаратного продукта и информационной системы для обработки документов;
* возможность работы с сервером;
* создание удобного поиска с применением технологии *RFID*;
* разработка интуитивного дизайна программного продукта.

Функции информационной системы:

* сбор информации;
* хранение информации;
* поиск и обработка информации;
* получение информации.

Для установки, разрабатываемой информационной программно-аппаратной системы необходим следующий состав технических средств:

* процессор не менее *Pentium-2.0Hz;*
* оперативную память объемом не менее 512 Мб;
* устройство ввода: клавиатура, мышь, *RFID* или сенсорный ввод;
* операционная система *Windows 7, Windows 8, Windows 8.1, Windows 10*;
* компоненты среды *Net. Framework 4.0*.

Входными данными служит электронная информация о документах.

Выходными данными являются результаты поиска.

**1.5 Выводы**

Поставленная задача является актуальной в сфере развития информационных технологий, применение в различных сферах человеческой деятельности, появление большого количества документов с чипом которые можно проверить бесконтактным методом и тем самым облегчить работу.

Была исследована поставленная задача и определен перечень функций, которые необходимо реализовать в программном продукте.

Основными пунктами рассмотрения в первой главе стали:

* описание поставленной задачи;
* обоснование актуальности;
* обзор методов решения подобных задач;
* постановка задачи, системные требования.

**2 РАЗРАБОТКА МЕТОДОВ РЕШЕНИЯ ПОСТАВЛЕННОЙ ЗАДАЧИ**

**2.1 Выбор методов и средств для реализации, его обоснование**

В настоящее время существует большой выбор средств разработки приложений на платформе операционной системы *Windows*. Ниже приведен обзор наиболее популярных средств.

*«***Microsoft SQL Server***» –* система управления реляционными базами данных, разработанная корпорацией *Microsoft* [12].

Профессиональная деятельность, так или иначе, связана с информацией, с организацией ее сбора, хранения, выборки. Можно сказать, что неотъемлемой частью повседневной жизни стали базы данных, для поддержки которых требуется некоторый организационный метод, или механизм. Такой механизм называется системой управления базами данных. Введем основные понятия.

База данных – совместно используемый набор логически связанных данных, предназначенный для удовлетворения информационных потребностей.

Система управления базами данных – программное обеспечение, с помощью которого пользователи могут определять, создавать и поддерживать базу данных, а также получать к ней контролируемый доступ.

Управление основными потоками информации осуществляется с помощью, так называемых, систем управления реляционными базами данных, которые берут свое начало в традиционных системах управления базами данных. Именно объединение реляционных баз данных и клиент-серверных технологий позволяет современному предприятию успешно управлять собственными данными, оставаясь конкурентоспособным на рынке товаров и услуг. Реляционные БД имеют мощный теоретический фундамент, основанный на математической теории отношений. Появление теории реляционных баз данных дало толчок к разработке ряда языков запросов, которые можно отнести к двум классам:

* алгебраические языки, позволяющие выражать запросы средствами специализированных операторов, применяемых к отношениям;
* языки исчисления предикатов, представляющие собой набор правил для записи выражения, определяющего новое отношение из заданной совокупности существующих отношений. Следовательно, исчисление предикатов есть метод определения того отношения, которое желательно получить как ответ на запрос из отношений, уже имеющихся в базе данных.

В реляционной модели объекты реального мира и взаимосвязи между ними представляются с помощью совокупности связанных между собой таблиц.

Даже в том случае, когда функции системы управления базами данных используются для выбора информации из одной или нескольких таблиц, результат также представляется в табличном виде. Более того, можно выполнить запрос с применением результатов другого запроса.

Каждая таблица базы данных представляется как совокупность строк и столбцов, где строки соответствуют экземпляру объекта, конкретному событию или явлению, а столбцы атрибутам объекта, события, явления.

В каждой таблице базы данных необходимо наличие первичного ключа так именуют поле или набор полей, однозначно идентифицирующий каждый экземпляр объекта или запись. Значение первичного ключа в таблице базы данных должно быть уникальным. Он должен быть минимально достаточным, а значит, не содержать полей, удаление которых не отразится на его уникальности.

Связи между объектами реального мира могут находить свое отражение в структуре данных, а могут и подразумеваться, то есть присутствовать на неформальном уровне.

Между двумя или более таблицами базы данных могут существовать отношения подчиненности, которые определяют, что для каждой записи главной таблицы возможно наличие одной или нескольких записей в подчиненной таблице.

Выделяют три разновидности связи между таблицами базы данных:

* «один-ко-многим»;
* «один-к-одному»;
* «многие-ко-многим».

Отношение «один-ко-многим» имеет место, когда одной записи родительской таблицы может соответствовать несколько записей дочерней.

Отношение «один-к-одному» имеет место, когда одной записи в родительской таблице соответствует одна запись в дочерней.

Отношение «многие-ко-многим» применяется в следующих случаях. Одной записи в родительской таблице соответствует более одной записи в дочерней. Или же одной записи в дочерней таблице соответствует более одной записи в родительской.

Дизайн программы изображен на рисунке 2.1.

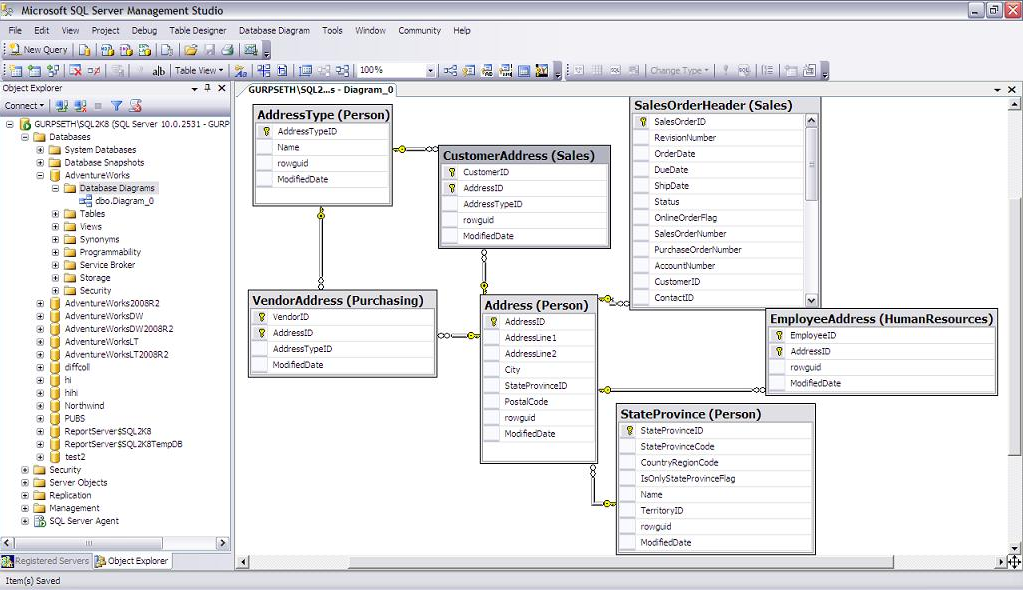


Рисунок 2.1 – Программа *Microsoft SQL Server*

Любая система управления базами данных позволяет выполнять четыре простейшие операции с данными:

* добавлять в таблицу одну или несколько записей;
* удалять из таблицы одну или несколько записей;
* обновлять значения некоторых полей в одной или нескольких записях;
* находить одну или несколько записей, удовлетворяющих условию.

Для выполнения этих операций используется механизм запросов. Результатом выполнения запросов является либо отобранное по определенным критериям множество записей из таблицы, либо изменения в таблицах. Запросы к базе формируются на специально созданном для этого языке, который так и называется – язык структурированных запросов.

Важной функцией системы управления базами данных является управление данными. Под этим обычно понимают защиту данных от несанкционированного доступа, поддержку многопользовательского режима работы с данными, а также обеспечение целостности и согласованности данных. Защита от несанкционированного доступа позволяет каждому пользователю видеть и изменять только те данные, которые ему разрешено видеть или менять. Средства, обеспечивающие многопользовательскую работу, не позволяют нескольким пользователям одновременно изменять одни и те же данные. Средства обеспечения целостности и согласованности данных не дают выполнять такие изменения, после которых данные могут оказаться несогласованными.

Выбор был остановлен на данном средстве реализации базы данных информационной системы исходя из всех вышеперечисленных качеств.

*«***Microsoft Visual Studio Enterprise 2015**» – идеально подходит для создания данного проекта, а именно автоматизированной информационной программно-аппаратной системы, имеет понятный интерфейс и большой набор функций, инструментов [2].

*Microsoft Visual Studio* – линейка продуктов компании *Microsoft*, включающих интегрированную среду разработки программного обеспечения и ряд других инструментальных средств. Данные продукты позволяют разрабатывать консольные приложения, приложения с графическим интерфейсом, в том числе с поддержкой технологии *Windows Forms*, а также веб-сайты, веб-приложения.

*Visual Studio* включает в себя редактор исходного кода с поддержкой технологии *IntelliSense* и возможностью простейшего рефакторинга кода. Встроенный отладчик может работать как отладчик уровня исходного кода, так и как отладчик машинного уровня. Остальные встраиваемые инструменты включают в себя редактор форм для упрощения создания графического интерфейса приложения, веб-редактор, дизайнер классов и дизайнер схемы базы данных. *Visual Studio* позволяет создавать и подключать сторонние дополнения (плагины) для расширения функциональности практически на каждом уровне, включая добавление поддержки систем контроля версий исходного кода, добавление новых наборов инструментов.

Каждая новая версия программы состоит из новейших инструментов и технологий, позволяющих разрабатывать приложения с учетом особенностей и положительных моментов современных платформ.

На рисунке 2.2 изображено начальное окно студии.

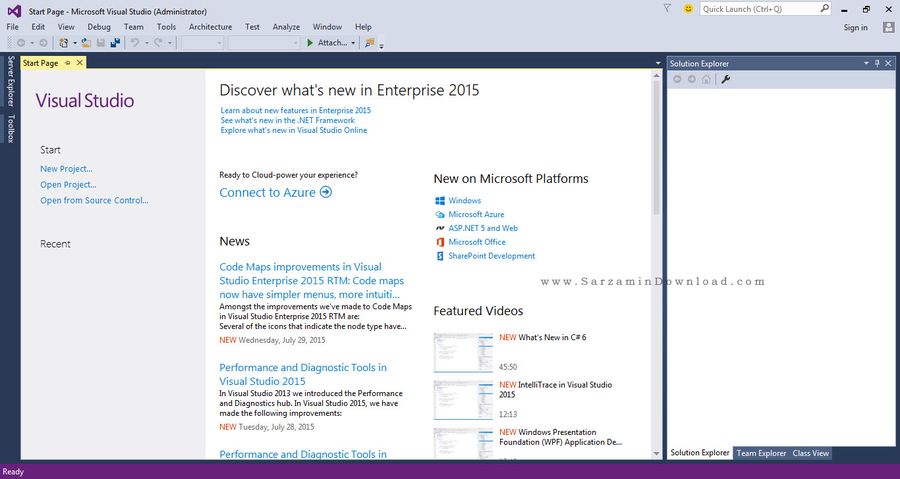


Рисунок 2.2 – Начальное окно *Microsoft Visual Studio Enterprise 2015*

Средства, входящие в состав *Visual* *Studio* для *Windows*, можно использовать для создания привлекательных инновационных приложений для Магазина *Windows* в среде *Windows* 8.1. Эти средства включают полнофункциональный редактор кода, мощный отладчик, специальный профилировщик и широкие возможности языковой поддержки, которые позволяют выполнять сборку приложений, написанных на языках *HTML*5/*JavaScript*, *C*++, *C*# и *Visual Basic*. В состав *Visual Studio* для *Windows* также входит имитатор устройств, который можно использовать для тестирования приложений магазина *Windows* на устройствах различных видов.

Данная среда программирования, позволяет работать с данными различного типа. Имеет понятный интерфейс, широкий набор инструментов для разработки приложений который расширяется. Интеграция со многими языками программирования и программами, мгновенно компилирует программный код и выявляет ошибки.

«**Arduino Uno**» – представляет собой инструмент, с помощью которого можно создавать различные электронные устройства. По сути, это аппаратная вычислительная платформа универсального предназначения. Можно использоваться как для построения простых схем, так и для реализации довольно сложных проектов. Базируется конструктор на своей аппаратной части, которая представляет собой плату ввода-вывода [5].

На рисунке 2.3 представлен дизайн программы для компилирования кода и заливки скетча в контроллерс простейшим скетчем моргания светодиода.

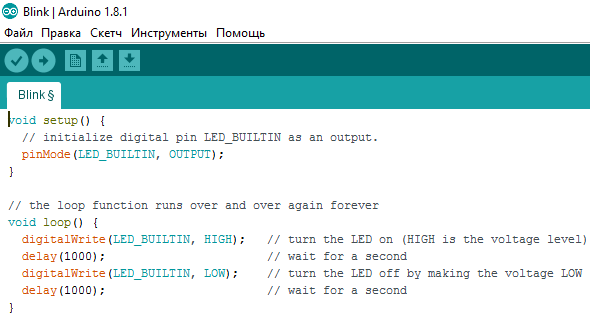


Рисунок 2.3 – Программа для работы с *Arduino*

Для программирования платы используются языки, которые основаны на *C/C*++. От группы *С* унаследовали предельную простоту, благодаря чему осваиваются весьма быстро любым человеком, и применять знания на практике не является довольно значительной проблемой.

Применение *Arduino* довольно разнообразно, можно использовать, как и для простейших примеров, так и для довольно сложных механизмов, среди которых манипуляторы, роботы или производственные станки. На основе таких систем возможно сделать планшет, телефон, системы наблюдения и безопасности домов, системы «умный дом» или просто компьютер.

На рисунке 2.4 представлен дизайн аппарата и описание разъемов.

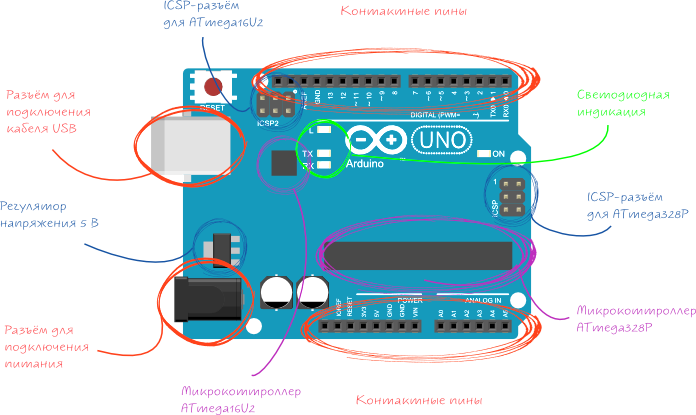


Рисунок 2.4 – Плата *Arduino Uno*

Преимущество плат *Arduino* перед аналогичными платформами относительно невысокая цена и практически массовое распространение среди любителей и профессионалов робототехники и электротехники. Имеет немало дополнительных модулей, с помощью которых можно собрать робота, сделать плеер, сконструировать автомобиль и т.д. Модули бывают разные такие как, двигатель с драйвером, инфракрасный датчик, радиомодуль, дисплей, видеокамера и т.д.

Готовность к использованию – является самым большим плюсом *Arduino*. Не нужно думать о программировании микроконтроллера или способах подключения периферии.

**2.2 Описание логической структуры решения задачи  
2**

Рассмотри данную задачу на примере таких документов как водительское удостоверение и технический паспорт транспорта. Данные документы обладают чипом которые позволяют применить данную *RFID* технологию.

Для построения информационной модели будущей системы, опишем информационные особенности предметной области, используя *ER*-модель. Это позволит представить информационные потребности в удобном виде.

Основными сущностями моделируемой области являются: «транспорт», «водительское удостоверение» и «протоколы». Для идентификации сущностей используется ключ, являющийся атрибутом сущности. Между сущностями «водительское удостоверение» и «протоколы» можно определить логическое отношение – один ко многим, поскольку одному водителю может принадлежать более одного протокола, а протокол, в свою очередь, может принадлежать лишь одному водителю. Для сущностей «протоколы» и «транспорт» логическое соотношение – многие к одному, поскольку транспорту может принадлежать более одного протокола, а протокол может принадлежать лишь одному транспорту.

Создадим базу данных в соответствии со следующими правилами:

* каждый атрибут сущности преобразуется в колонку таблицы, которая называется полем;
* имя атрибута становится именем поля;
* каждая строка таблицы является экземпляром сущности;
* ключевые атрибуты становятся первичными ключами таблиц;
* если для сущности была определена ключевая связь, то первичный ключ связанной сущности копируется в таблицу и объединяется с ключом таблицы рассматриваемой сущности.

Требуется представить физическую модель информационной базы данных с сущностями «водительское», «транспорт», «протоколы», а также каждая таблица заполнена атрибутами.

На рисунке 2.5 представлена физическая модель информационной базы данных с сущностями.

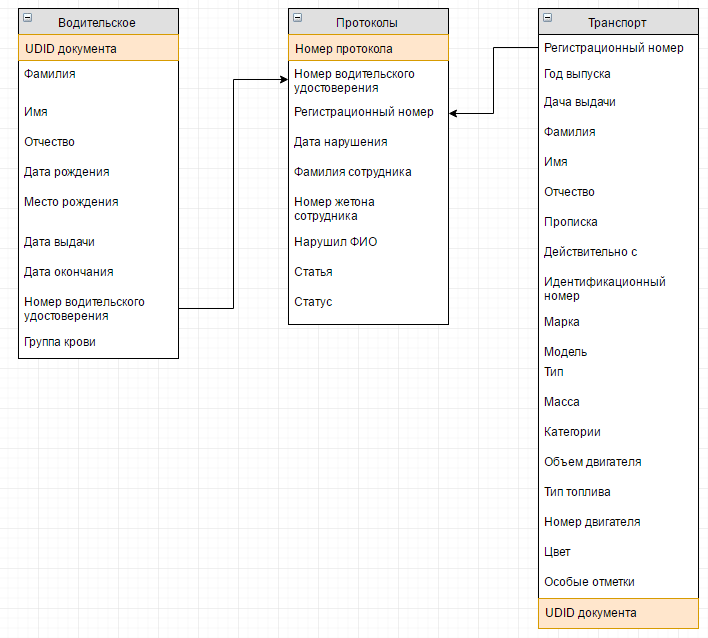


Рисунок 2.5 – Диаграмма базы данных информационной системы

Рассмотрим алгоритм формирования списка штрафов. Для этого необходимо произвести выборку данных из таблиц *«*номер документа*», «*фамилия*»,* «регистрационный номер», «номер протокола», «дата нарушения», «фамилия сотрудника», «номер жетона сотрудника», «нарушил ФИО», «статья», «статус» из базы данных *«database»* по средствам *SQL* запроса. Полученные данные будут помещены в объект *info*, который выступает источником данных для полей данной формы на рисунке 2.6.

Помещение выбранных

записей в *dbAdapter*

Формирование

*info*

Формирование *SQL*

запроса к базе данных

Рисунок 2.6 – Формирование запроса

После формированного *SQL* запроса к базе данных, будет сформирован вывод по выбраным столбцам и таблицам.

Представим графическое представление *ER*–модели предметной области. На рисунке 2.7 предсавтлена графическая диаграмма.



Рисунок 2.7 – *ER* модель предметной области

Исходя из описания предметной области, для каждой сущности можно определить набор атрибутов. Атрибуты и их имена представлены в таблице 2.1.

Таблица 2.1 – Набор атрибутов

|  |  |  |
| --- | --- | --- |
| **Водительское** | **Транспорт** | **Протоколы** |
| *UDID* документа | Регистрационный номер | Номер протокола |
| Фамилия | Год выпуска | Номер водительского  удостоверения |
| Имя | Дата выдачи | Регистрационный номер |
| Отчество | Фамилия | Дата нарушения |
| Дата рождения | Имя | Фамилия сотрудника |
| Место рождения | Отчество | Номер жетона сотрудника |
| Дата выдачи | Прописка | Нарушил ФИО |
| Дата окончания | Действительно с | Статья |
| Номер водительского  удостоверения | Идентификационный номер | Статус |
| Группа крови | Марка |  |
|  | Модель |  |
|  | Тип |  |
|  | Масса |  |
|  | Категории |  |
|  | Продолжение таблицы 2.1 |  |
|  | Объем двигателя |  |
|  | Тип топлива |  |
|  | Номер двигателя |  |
|  | Особые отметки |  |
|  | *UDID* документа |  |

Дополненная *ER* модель наборами атрибутов представляет собой концептуальную модель предметно области.

Данную концептуальную модель мы можем рассмотреть на рисунке 2.8.

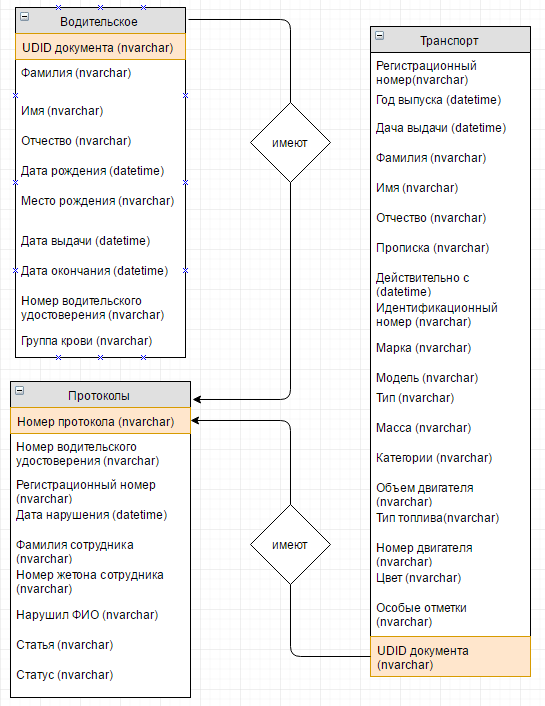


Рисунок 2.8 – Концептуальная модель

На основе анализа *ER*–диаграммы предметной области, можно составить следующие отношения.

Отношение «Водительское удостоверение» обладает следующими атрибутами: *UDID* документа, Фамилия, Имя, Отчество, Дата рождения, Место рождения, Дата выдачи, Дата окончания, Номер водительского удостоверения, Группа крови. Атрибуты представлены ниже.

Таблица 2.2 – Таблица «Водительское удостоверение»

|  |  |  |  |
| --- | --- | --- | --- |
| **Имя атрибута** | **Имя поля** | **Тип данных** | **Описание** |
| *UDID* документа | *UDID* | Текстовый | Содержит *UDID* документа. Данный атрибут  является идентифицирующим и уникальным для каждой из записей |
| Фамилия | *familia* | Текстовый | Атрибут содержит  фамилию водителя. |
| Имя | *name* | Текстовый | Атрибут содержит имя водителя. |
| Отчество | *otchestvo* | Текстовый | Атрибут содержит  отчество водителя. |
| Дата рождения | *datar* | Дата/Время | Атрибут содержит дату рождения. |
| Место рождения | *mestor* | Дата/Время | Атрибут содержит  место рождения. |
| Дата выдачи | *datav* | Дата/Время | Атрибут содержит  дату выдачи прав |
| Дата окончания | *datao* | Дата/Время | Атрибут содержит дату окончания действия прав |
| Номер водительского удостоверения | *serianomer* | Текстовый | Атрибут содержит  номер удостоверения |

Отношение «протоколы» обладает следующими атрибутами: Номер протокола, Номер водительского удостоверения, Регистрационный номер, Дата нарушения, Фамилия сотрудника, Номер жетона сотрудника, Нарушил ФИО, Статья, Статус. Атрибуты представлены ниже.

Таблица 2.3 – Таблица «Протоколы»

|  |  |  |  |
| --- | --- | --- | --- |
| **Имя атрибута** | **Имя поля** | **Тип данных** | **Описание** |
| Номер  протокола | *nomerprotocola* | Текстовый | Атрибут содержит номер протокола и является идентифицирующим и уникальным для каждой из записей |
| Номер  водительского удостоверения | *nomervod* | Текстовый | Атрибут содержит номер  водительского удостоверения. |
| Регистрационный номер | *registranomer* | Текстовый | Атрибут содержит номер  транспорта. |
| Дата нарушения | *datanarush* | Дата/Время | Атрибут содержит дату нарушения. |
| Фамилия  сотрудника | *famsotr* | Текстовый | Атрибут содержит фамилию  сотрудника. |
| Номер жетона сотрудника | *nomerwetonasotr* | Текстовый | Атрибут содержит номер  сотрудника. |
| Нарушил ФИО | *narfio* | Текстовый | Атрибут содержит данные  нарушителя. |
| Статья | *statia* | Текстовый | Атрибут содержит номер статьи. |
| Статус | *status* | Текстовый | Атрибут содержит статус оплаты. |

Отношение «транспорт» обладает следующими атрибутами: Регистрационный номер, Год выпуска, Дата выдачи, Фамилия, Имя, Отчество, Прописка, Действительно с, Идентификационный номер, Марка, Модель, Тип, Масса, Категории, Объем двигателя, Номер двигателя, Особые отметки, *UDID* документа. Данные атрибуты можно рассмотреть в таблице 2.4.

Таблица 2.4 – Таблица «Транспорт»

|  |  |  |  |
| --- | --- | --- | --- |
| **Имя атрибута** | **Имя поля** | **Тип данных** | **Описание** |
| Регистрационный номер | *transportnomer* | Текстовый | Атрибут содержит  регистрационный  номер. |
| Год выпуска | *godvipuska* | Дата/Время | Атрибут содержит год выпуска. |
| Дата выдачи | *datavidac4i* | Дата/Время | Атрибут содержит дату выдачи. |
| Фамилия | *familiaa* | Текстовый | Атрибут содержит  фамилию владельца. |
| Имя | *imia* | Текстовый | Атрибут содержит имя владельца. |
| Отчество | *ot4estvo* | Текстовый | Атрибут содержит  отчество владельца. |
| Прописка | *adrespropiska* | Текстовый | Атрибут содержит прописку владельца |
| Действительно с | *deistvitelyno* | Дата/Время | Атрибут содержит дату выдачи |
|  | Продолжение таблицы | 2.4 |  |
| Идентификационный  номер | *identifinomer* | Текстовый | Атрибут содержит уникальный номер |
| Марка | *marka* | Текстовый | Атрибут содержит марку авто |
| Модель | *modely* | Текстовый | Атрибут содержит  модель авто |
| Тип | *tip* | Текстовый | Атрибут содержит тип авто |
| Масса | *massa* | Текстовый | Атрибут содержит массу авто |
| Категории | *kategorii* | Текстовый | Атрибут содержит  категории |
| Объем двигателя | *obyemdvigatelya* | Текстовый | Атрибут содержит объем двигателя |
| Тип топлива | *tiptopliva* | Текстовый | Атрибут содержит тип топлива |
| Номер двигателя | *nomerdvigatelya* | Текстовый | Атрибут содержит  номер двигателя |
| Цвет | *cvet* | Текстовый | Атрибут содержит цвет авто |
| Особые отметки | *osobieotmetki* | Текстовый | Атрибут содержит особые отметки |
| *UDID* документа | *udidtransporta* | Текстовый | Атрибут содержит номер документа. Данный атрибут является идентифицирующим и уникальным для каждой из записей |

**2.3 Функциональная схема решения задачи**

Пользовательский интерфейс является неотъемлемой частью большинства систем. От качества интерфейса во многом зависит эффективность использования конкретной системы. Исследования в этой области показывают, что в среднем около 48% программного кода приложений относится к реализации пользовательского интерфейса, при этом, чем проще становится интерфейс, тем сложнее оказывается процесс его разработки.

Пользователь воспринимает через пользовательский интерфейс всю систему в целом. В действительности пользовательский интерфейс включает в себя все аспекты дизайна, которые оказывают влияние на взаимодействие пользователя и системы. Это не только экран, который видит пользователь.

Пользовательский интерфейс состоит из множества составляющих:

* набор задач пользователя, которые он решает при помощи системы;
* элементы управления системой;
* навигация между блоками системы;
* визуальный дизайн экранов программы.

На рисунке 2.9 представлена функциональная схема.

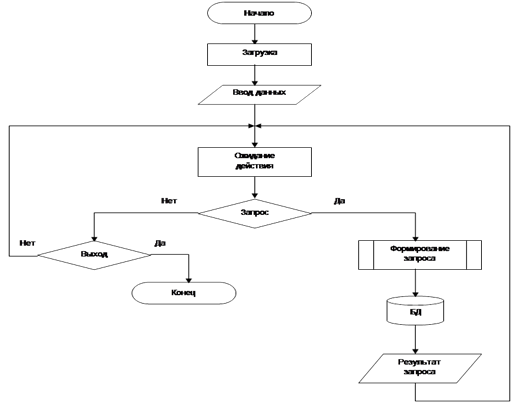
  
2

Рисунок 2.9 – Функциональная схема

При разработке был применен итерационный процесс проектирования интерфейса пользователя информационной системы.   Использовались следующие принципы проектирования пользовательских интерфейсов.

*Учет знаний пользователя.* В интерфейсе необходимо использовать термины и понятия, взятые из опыта будущих пользователей системы.

*Согласованность.* Интерфейс информационной системы должен быть согласованным в том смысле, что однотипные операции должны выполняться одним и тем же способом.

*Минимум неожиданностей.* Поведение информационной системы должно быть прогнозируемым.

*Руководство пользователя.* Интерфейс должен предоставлять необходимую информацию в случае ошибок пользователя и поддерживать средства контекстно-зависимой справки.

*Удобство.* Легкость, простота, надежность и продуктивность в использовании интерфейса.

Интерфейс главного окна представлен на рисунке 2.10.

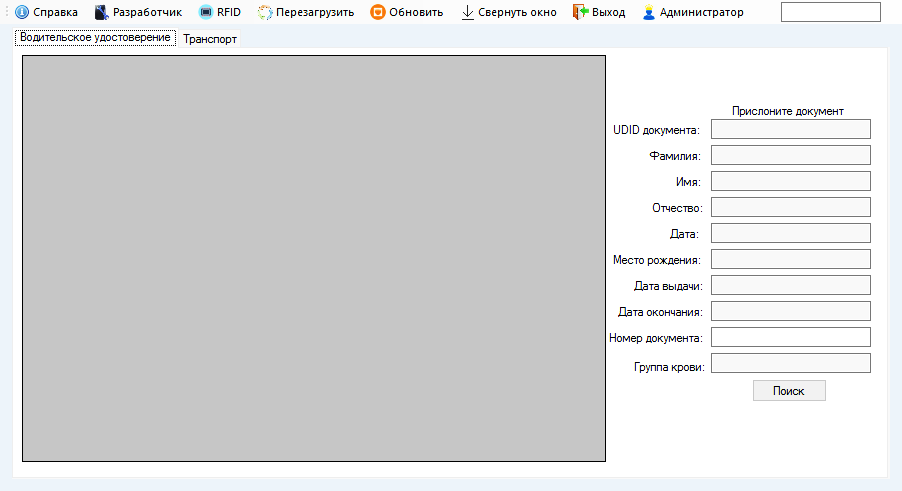


Рисунок 2.10 – Главное окно программного продукта

Данная программа позволяет работать с информационной базой данных и использовать устройство радиочастотной идентификации документа.

**2.4 Выводы**

Во второй главе выпускной квалификационной работы были выбраны средства разработки, а именно *Microsoft Visual Studio Enteprise* 2015, *SQL*, *Arduino*. Были описаны основные методы реализации программного продукта и его структура. Реализованы диаграммы предметной области, концептуальная модель и описан набор атрибутов. Так же была описана функциональная схема программного продукта.

Основными пунктами рассмотрения во второй главе стали:

* выбор методов и средств для реализации, его обоснование;
* структура, архитектура программного продукта;
* функциональная схема, функционал программного продукта.

**3 РЕАЛИЗАЦИЯ И ТЕСТИРОВАНИЕ ВЫБРАННЫХ МЕТОДОВ**

**РЕШЕНИЯ ЗАДАЧ  
2**

**3.1 Описание реализации  
3**

Разрабатываемый программно-аппаратный продукт состоит из трех частей. Первой частью является информационная база данных которая позволяет хранить в себе информацию и взаимодействовать с ней. Вторая часть – это аппарат, работающий на микроконтроллере, который отвечает за радиочастотное считывание и взаимодействия с программным продуктом. Третья часть является программа, с помощью которой возможно получить всю нужную информацию в данном направлении, управлять базой данных и аппаратной частью.

Рассмотрим первую часть разработки информационной базы данных. Для создания базы данных написали запрос.

CREATE DATABASE database

Листинг 3.1 – Создание БД

Для хранения и взаимодействия с информацией также требуется создать несколько таблиц. Рассмотрим синтаксис таблицы «*protocol*» в которой хранится информация о нарушителях и статуса оплаты штрафа в реальном времени. По полю «*nomerprotocola*» используется ограничение *UNIQUE*. При попытке добавить повторяющиеся значения, выдаст ошибку.

CREATE TABLE [dbo].[protocol] (

[nomerprotocola] NVARCHAR (50) NULL,

[nomervod] NVARCHAR (50) NULL,

[registranomer] NVARCHAR (50) NULL,

[datanarush] DATE NULL,

[famsotr] NVARCHAR (50) NULL,

[nomerwetonasotr] NVARCHAR (50) NULL,

[narfio] NVARCHAR (50) NULL,

[statia] NVARCHAR (50) NULL,

[status] NVARCHAR (50) NULL,

UNIQUE NONCLUSTERED ([nomerprotocola] ASC));

Листинг 3.2 – Создание таблицы протоколы

Рассмотри таблицу «*transport*» которая хранит в себе данные транспортного средства такие как регистрационный номер, цвет, собственник, идентификационный номер и т.д. Используется один и тот же тип данных во всех таблицах для работы с датой и текстом.

CREATE TABLE [dbo].[transport] (

[transportnomer] NVARCHAR (50) NULL,

[godvipuska] DATE NULL,

[datavidac4i] DATE NULL,

[familiaa] NVARCHAR (50) NULL,

[imia] NVARCHAR (50) NULL,

[ot4estvoo] NVARCHAR (50) NULL,

[adrespropiska] NVARCHAR (50) NULL,

[deistvitelyno] DATE NULL,

[identifinomer] NVARCHAR (50) NULL,

[marka] NVARCHAR (50) NULL,

[modely] NVARCHAR (50) NULL,

[tip] NVARCHAR (50) NULL,

[massa] NVARCHAR (50) NULL,

[kategorii] NVARCHAR (50) NULL,

[obyemdvigatelya] NVARCHAR (50) NULL,

[tiptopliva] NVARCHAR (50) NULL,

[nomerdvigatelya] NVARCHAR (50) NULL,

[cvet] NVARCHAR (50) NULL,

[osobieotmetki] NVARCHAR (50) NULL,

[udidtransport] NVARCHAR (50) NULL,

UNIQUE NONCLUSTERED ([udidtransport] ASC)

);

Листинг 3.3 – Создание таблицы транспорт

Так же создали таблицу «*VU*» для хранения данных водительского удостоверения. Синтаксис представлен ниже.

CREATE TABLE [dbo].[VU] (

[UDID] NVARCHAR (50) NULL,

[familia] NVARCHAR (50) NULL,

[name] NVARCHAR (50) NULL,

[otchestvo] NVARCHAR (50) NULL,

[datar] DATE NULL,

[mestor] NVARCHAR (50) NULL,

[datav] DATE NULL,

[datao] DATE NULL,

[serianomer] NVARCHAR (50) NULL,

[gruppakrovi] NVARCHAR (50) NULL,

UNIQUE NONCLUSTERED ([UDID] ASC));

Листинг 3.4 – Создание таблицы водительское удостоверение

Вторая часть – это принцип работы аппарата. Написали программный код для заливки в микроконтроллер, который управляет модулем радиочастотного считывания и модулем для передачи данных с помощью *bluetooth.* Ниже представлен скетч для работы микроконтроллера.

#include <SPI.h>

#include <MFRC522.h>

#define RST\_PIN 9

#define SS\_PIN 10

MFRC522 mfrc522(SS\_PIN, RST\_PIN);

MFRC522::MIFARE\_Key key;

int buzzPin = 3;

void setup() {

Serial.begin(9600);

SPI.begin();

mfrc522.PCD\_Init();

}

void loop() {

if ( ! mfrc522.PICC\_IsNewCardPresent())

return;

if ( ! mfrc522.PICC\_ReadCardSerial())

return;

dump\_byte\_array(mfrc522.uid.uidByte, mfrc522.uid.size);

mfrc522.PICC\_HaltA();

mfrc522.PCD\_StopCrypto1();

}

void dump\_byte\_array(byte \*buffer, byte bufferSize) {

for (byte i = 0; i < bufferSize; i++) {

Serial.print(buffer[i] < 0x10 ? "0" : "");

Serial.print(buffer[i], HEX);

analogWrite(3,500);

delay(50);

analogWrite(3,0);

}

Serial.println();

}

Листинг 3.5 – Код для работы контроллера

База данных создана, аппаратная часть собрана. Чтобы заработали обе части требуется третья часть по созданию программного продукта.

Для работы аппаратной и информационной базы данных требуется подключить нужные библиотеки. При подключении базы данных используется строка подключения, представленная ниже.

public string connStr = @"Data Source

(LocalDB)\MSSQLLocalDB;AttachDbFilename=C:\Users\Asssss\Desktop\RFID\RFID\database

df;Integrated Security=True;";

Листинг 3.6 – Подключение БД

Для аппаратной части требуется порт, по которому будут передаваться данные за счет бесконтактной технологии.

SerialPort port = new SerialPort();

port.PortName = Properties.Settings.Default.PortName;

port.Open();

port.Close();

Листинг 3.7 – Проверка и открытие порта

При работе с базой данных требуется открыть соединение, выполнить запрос, вывести информацию в элемент *dataGridView* и закрыть соединение.

SqlConnection conn = new SqlConnection(connStr);

Conn.Open();

string SQLstr = "SELECT \* FROM VU";

SqlDataAdapter da = new SqlDataAdapter(SQLstr, conn);

DataTable t = new DataTable("VU");

da.Fill(t);

dataGridView1.DataSource = t;

conn.Close();

Листинг 3.8 – Запрос таблицы «водительское удостоверение»

Информационная база данных позволяет по запросу с легкостью найти нужный документ. Запрос приведен ниже.

SqlCommand cmd = new SqlCommand("SELECT \* FROM VU WHERE UDID ='" + UDIDBox.Text + "'", conn);

Листинг 3.9 – Поиск документа

Чтобы добавить информацию в базу данных открываем соединение, составляем запрос, в котором укажем цель и нужные атрибуты. Ниже приведен пример для добавления *UDID* номера.

SqlConnection conn = new SqlConnection(connStr);

conn.Open();

SqlCommand cmd = new SqlCommand("Insert into VU ([UDID], [familia], [name],

[otchestvo], [datar], [mestor], [datav],

[datao], [serianomer], [gruppakrovi]) Values

(@param0,@param1,@param2,@param3,@param4,@param5,@param6,@param7,@param8,@param9)"

, conn);

SqlParameter param;

param = new SqlParameter();

param.ParameterName = "@param0";

param.Value = textBox12.Text;

param.SqlDbType = SqlDbType.NVarChar;

cmd.Parameters.Add(param);

conn.Close();

conn.Dispose();

Листинг 3.10 – Добавление данных в БД

Для редактирования данных в базе требуется учесть какой именно атрибут нужно отредактировать. Возможно редактировать как все строки, так и одну. Приведем пример кода.

SqlConnection conn = new SqlConnection(connStr);

conn.Open();

SqlCommand cmdupd = new SqlCommand ("UPDATE [VU] " + " SET [familia]=@s2,

[name]=@s3, [otchestvo]=@s4, [datar]=@s5, [mestor]=@s6, [datav]=@s7, [datao]=@s8,

[serianomer]=@s9, [gruppakrovi]=@s10 " + " WHERE [UDID]=@s1", conn);

cmdupd.Parameters.AddWithValue("@s1", textBox12.Text);

cmdupd.Parameters.AddWithValue("@s2", textBox11.Text);

cmdupd.Parameters.AddWithValue("@s3", textBox10.Text);

cmdupd.Parameters.AddWithValue("@s4", textBox9.Text);

cmdupd.Parameters.AddWithValue("@s5", textBox8.Text);

cmdupd.Parameters.AddWithValue("@s6", textBox7.Text);

cmdupd.Parameters.AddWithValue("@s7", textBox6.Text);

cmdupd.Parameters.AddWithValue("@s8", textBox5.Text);

cmdupd.Parameters.AddWithValue("@s9", textBox4.Text);

cmdupd.Parameters.AddWithValue("@s10", textBox2.Text);

conn.Close();

conn.Dispose();

Листинг 3.11 – Редактирование данных

Со временем некоторая информация, хранящаяся в базе данных, устаревает и теряет свою актуальность. Поэтому требуется реализовать так же возможность удаления ненужной информации.

SqlConnection conn = new SqlConnection(connStr);

conn.Open();

SqlCommand cmd22 = new SqlCommand("DELETE FROM VU WHERE UDID= '" + textBox1.Text +

"'", conn);

conn.Close();

conn.Dispose();

Листинг 3.12 – Удаление данных

Информационный программно-аппаратный продукт полностью выполнен и готов к работе.

**3.2 Описание пользовательского интерфейса**

Главное окно программного продукта представлен на рисунке 3.1.

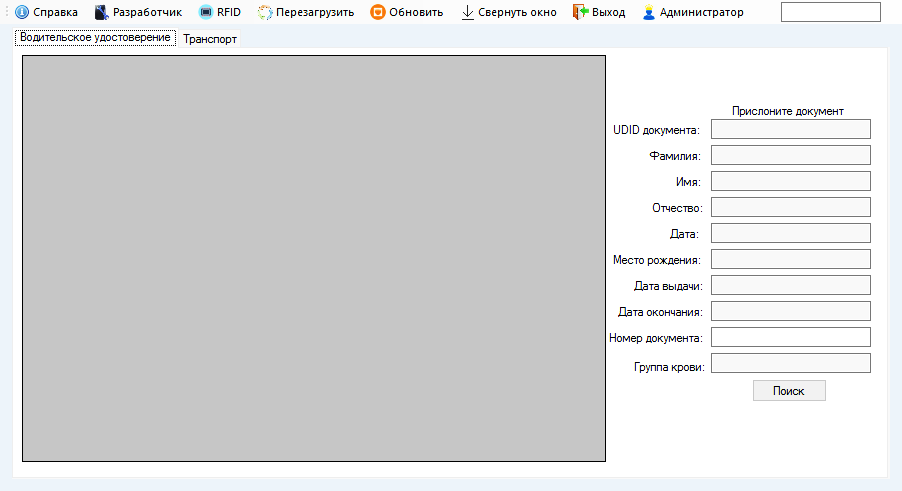


Рисунок 3.1 – Главное окно программного продукта

На главном окне присутствуют такие элементы как «справка – информация по работе с программой», «разработчик – данные о разработчике», «*RFID* – настройки по выбору порта для работы устройства как дистанционно так и стационарно», «перезагрузить – позволяет перезапустить приложение», «обновить – позволяет обновить данные базы данных и подгрузить последние изменения», «свернуть окно – позволяет свернуть окно в трей», «выход – закрывает программу, соединение с базой данных и аппаратом», «администратор – при вводе пароля переходим в привилегированный режим который позволяет обрабатывать информацию».

Для поиска информации гражданина и транспорта достаточно две вкладки. В данных вкладах разрешено только искать информацию и не более. Рассмотреть можно на рисунке 3.2.

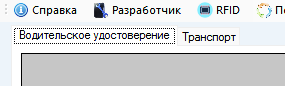


Рисунок 3.2 – Поиск информации

При вводе пароля отображаются дополнительные три вкладки, предназначенные для администратора, которые позволяют обрабатывать информацию. Представлено на рисунке 3.3.

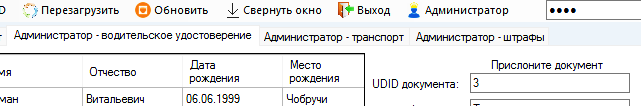


Рисунок 3.3 – Привилегированный режим

Поиск позволяет распознать документ бесконтактной технологией, что определяет подлинность документа. Так же выводится полностью информация в соответствии с документом и дополнительный список с нарушениями и статусом оплаты на рисунке 3.4.

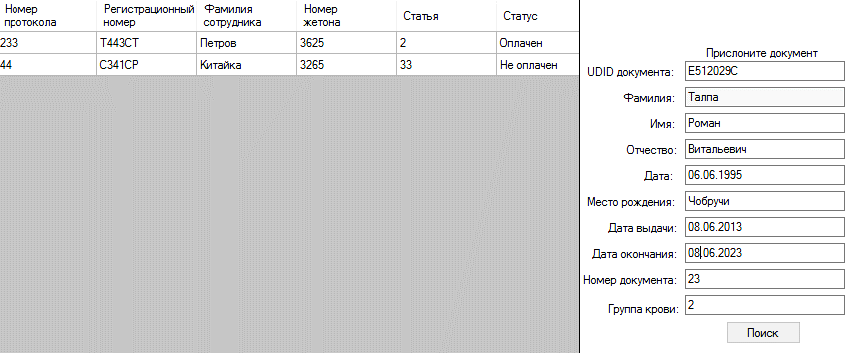


Рисунок 3.4 – Поиск документа

Для администратора поиск, редактирование, добавление и удаление представлено на рисунке 3.5.

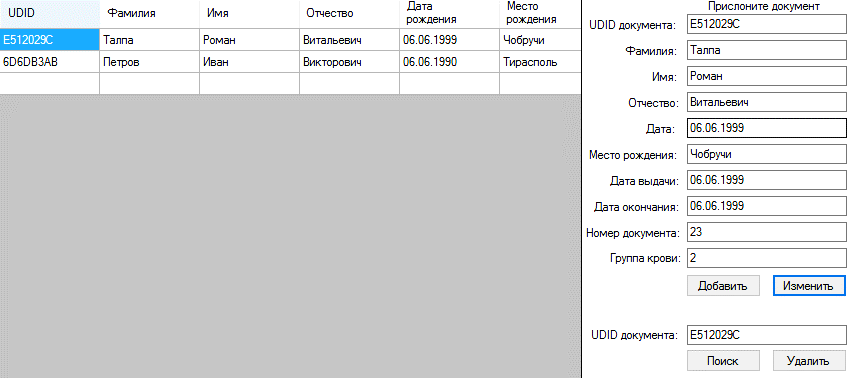


Рисунок 3.5 – Режим администратора

Дизайн, расположение элементов и их значение описано. Программа проста дизайном и принципом работы.

**3.3 Методы и средства защиты метода решения задачи**

Программные продукты и компьютерные базы данных являются предметом интеллектуального труда специалистов высокой квалификации. Процесс проектирования и реализации программных продуктов характеризуется значительными материальными и трудовыми затратами, основан на использовании наукоемких технологий и инструментария, требует применения и соответствующего уровня дорогостоящей вычислительной техники. Это обусловливает необходимость принятия мер по защите интересов разработчика программ и создателей компьютерных баз данных от несанкционированного их использования.

Программное обеспечение является объектом защиты также и в связи со сложностью и трудоемкостью восстановления его работоспособности, значимостью программного обеспечения для работы информационной системы.

Защита программного обеспечения преследует цели:

* ограничение несанкционированного доступа к программам или их преднамеренное разрушение и хищение;
* исключение несанкционированного копирования программ.

Программный продукт и базы данных должны быть защищены по нескольким направлениям от воздействия:

1. человека – хищение машинных носителей и документации программного обеспечения; нарушение работоспособности программного продукта и др.;
2. аппаратуры – подключение к компьютеру аппаратных средств для считывания программ и данных или их физического разрушения;
3. специализированных программ – приведение программного продукта или базы данных в неработоспособное состояние (например, вирусное заражение), несанкционированное копирование программ и базы данных и т.д.

Самый простой и доступный способ защиты программных продуктов и базы данных ограничение доступа. Контроль доступа к программному продукту и базе данных строится путем:

* парольной зашиты программ при их запуске;
* использования ключевой дискеты для запуска программ;
* ограничения программ или данных, функций обработки.

Было решено использовать парольную защиту. Так же в будущем требуется реализовать возможность доступа к правам администратора и базы данных с помощью бесконтактной технологии. Это позволит максимально защитить программу, не считая человеческого фактора.

**3.4 Тестирование и оценка надежности программного продукта**

При тестировании программного-аппаратного продукта были выявлены следующие ошибки:

* ввод различных символов;
* доступ к привилегированному режиму;
* пустое поле;
* бесконтактное считывание с лишним значением.

1) Ошибка с вводом различных символов была решена следующим методом. Каждое поле проверяется на вводимое значение, либо цифры, либо буквы. У одного поле может быть только кириллица, у другого латиница. Проблема с вводом цифр была решена добавлением кода в событие.

1

KeyPress: «e.Handled=!System.Text.RegularExpressions.Regex.IsMatch(e.KeyChar.ToString(), @"[0-9\b]");».

Листинг 3.13 – Проверка вводимого значения

Проверка на вводимые буквы:

1

«char l = e.KeyChar; if ((l < 'А' || l > 'я') &&l != '\b') {e.Handled = true;}».

Листинг 3.14 – Проверка вводимого значения

2) Доступ к привилегированному режиму. Так как не каждому сотруднику разрешен данный режим, было решено использовать пароль для входа.

3) Пустое поле. При поиске программа сообщает что поле пустое и нужно ввести значение. Код проверки пустого поля

1

«if (UDIDBox.Text != "") {} else MessageBox.Show(“Поле пустое”);».

Листинг 3.15 – Проверка поля

4) При считывании документа в переменную отправлялся дополнительный невидимый символ. Удалось распознать причину за счет средств диагностики в разрабатываемом программном продукте. Проблема была решена следующим методом

5

«entrada = entrada.Split('\r')[0];».

Листинг 3.16 – Удаление невидимого символа

**3.5 Расчет себестоимости от внедрения результатов выпускной   
квалификационной работы бакалавра**

Показатель экономической эффективности использования программных средств (ПС) учитывается при обосновании плановых показателей эффективности применения вычислительной техники, таких как экономия от снижения себестоимости продукции, увеличение объема производства, и др. Определение эффективности ПС основано на принципах определения экономической эффективности производства и использования новой техники [15].

Экономический эффект на стадии использования ПС рассчитывается по показателям замены используемых ПС более прогрессивным или по показателям включения в систему нового ПС с целью автоматизации работы.

На различных стадиях жизненного цикла ПС и в зависимости от цели рассчитываются и документально оформляются следующие виды экономического эффекта: предварительный; потенциальный; гарантированный; фактический.

Предварительный экономический эффект рассчитывается до выполнения разработки на основе данных технических предложений. Потенциальный экономический эффект рассчитывается по окончании разработки на основе достигнутых технико-экономических характеристик и прогнозных данных о максимальных объемах использования ПС в народном хозяйстве, используется при оценке деятельности организации разработчиков ПС. Гарантированный экономический эффект рассчитывается в виде гарантированного экономического эффекта для конкретного объекта внедрения и общего гарантированного внедрения по ряду объектов, рассчитывается после окончания разработки для одного программного внедрения на основе данных о гарантированном разработчиком удельном эффекте от применения ПС. Фактический экономический эффект рассчитывается на основе данных учета и сопоставления затрат и результатов при конкретных применениях ПС и рассчитывается от одного программного внедрения конкретного ПС на конкретном объекте, а также как общий экономический эффект от использования конкретного ПС на всех объектах внедрения за расчетный период. Фактический эффект используется для оценки деятельности организаций, разрабатывающих, внедряющих и использующих ПС, для определения отчислений в фонды экономического стимулирования, а также для анализа эффективности функционирования ПС и выработки технических предложений по усовершенствованию ПС и условий его применения.

Основными источниками экономии для предприятий, являются:

* улучшение показателей основной деятельности, при использовании ПС;
* сокращение сроков освоения новых ПС;
* сокращение расхода машинного времени и других ресурсов на отладку и сдачу задач в эксплуатацию;
* повышение уровня качества и объемов вычислительных работ;
* увеличение объемов и сокращение переработки информации;
* уменьшение численности персонала, в том числе высококвалифицированного, занятого обслуживанием программных средств;
* снижение трудоемкости работ программистов при программировании прикладных задач с использованием новых ПС.

3.5.1 Расчет затрат на проектирование и внедрение

Из всех выше перечисленных расчетов экономической эффективности ПС наиболее приемлемым является расчет гарантированного экономического эффекта для конкретного объекта внедрения. В структуре капитальных вложений, связанных со статистической обработкой, выделяют капитальные вложения на разработку проекта и капитальные вложения на реализацию проекта:

(3.1)

где – *Кп* – капитальные вложения на проектирование; *Кр* – капитальные вложения на реализацию проекта.

**3.5.2 Расчет капитальных вложений на проектирование**

Капитальные вложения на проектирование ПС определяются путем составления смет расходов и включают следующие элементы: стоимость материа-

лов (*Км*), заработную плату основную и дополнительную с отчислениями в соцстрах инженерно–технического персонала, непосредственно занятого разработкой проекта (*Кпр*); затраты, связанные с использованием машинного времени на отладку программы (*Кмаш*); оплату услуг сторонним организациям (*Кс*), если проектирование производится с привлечением сторонних организаций; накладные расходы отдела проектирования (*Кн*).

Таким образом, капитальные вложения на проектирование (*Кп*) определяются по формуле:

(3.2)

Все расчеты будут производиться в условных единицах (у.е.), что соответствует стоимости одного доллара США в Приднестровском Республиканском Банке на момент разработки ПС.

**3.5.3 Затраты на материалы**

Определим смету затрат и рассчитаем стоимость материалов *Км* пошедших на разработку ПС. В их состав входят материалы, отраженные в смете затрат представлены в таблице 3.1.

Таблица 3.1 – Смета затрат на материалы

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Материал | Единица изм. | Цена за ед., (у.е.) | Кол-во | Сумма, (у.е.) |
| Корпус | шт. | 2 | 1 | 2 |
| Драйвер заряда | шт. | 1 | 1 | 1 |
| Аккумулятор | шт. | 4 | 1 | 4 |
| Спикер | шт. | 0,5 | 1 | 0,5 |
| *Bluetooth -*модуль | шт. | 4 | 1 | 4 |
| *RFID-RC522 модуль* | шт. | 3 | 1 | 3 |
| Пластиковые метки | шт. | 0,4 | 10 | 4 |
| Микроконтроллер | шт. | 1 | 1 | 1 |
| Программатор | шт. | 1,5 | 1 | 1,5 |
| Кнопка включения | шт. | 0,3 | 1 | 0,3 |
| Перемычки | шт. | 0,05 | 11 | 0,55 |
| *USB* - провод | шт. | 1 | 1 | 1 |

Итого: 21,95 у.е. Транспортно–заготовительные расходы (5 %): 1,1 у.е.

Всего: 23,05 у.е.

**3.5.4 Затраты на оплату труда**

Затраты на основную заработную плату программиста (*Kпр*) рассчитываются на основе данных о квалификационном составе разработчиков, их должностных окладах и общей занятости по теме.

Дополнительная заработная плата начисляется в размере 10 % от суммы основной заработной платы, а отчисления на социальные страхования – в размере 39 % от фонда заработной платы. Смета затрат на оплату труда представлена в таблице 3.2.

Таблица 3.2 – Смета затрат на оплату труда.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Должность  работника | Должностной оклад (у.е.) | Дневная ставка | Занятость по теме | Сумма основной з/п (у.е.) |
| Программист | 50 | 1,818 | 70 | 127 |

Итого *Kпр*: 127 у.е.

3.5.5 Затраты на отладку программы

Затраты, связанные с использованием машинного времени на отладку программ (*Кмаш*), учитываются для следующих этапов проектирования: разработка рабочего проекта; внедрение – проведение опытной эксплуатации задач и сдача их в эксплуатацию.

Затраты на отладку программы определяются по формуле:

(3.3)

где – *Смч* – стоимость одного часа машинного времени; *Тотп* – время отладки программы (ч); *Sпр* – количество программистов.

Подставляя данные, получаем величину затрат на отладку программ:

*Смч* = 0,1 у.е.; *Тотл* = 60 ч.; *Sпр* = 1 программист, Кмаш = 0,1\*60\*1=6 у.е.

3.5.6 Оплата сторонним организациям и накладные расходы

Затраты на оплату работ сторонним организациям не производились, поэтому коэффициент *Кс*=0. Накладные расходы (*Кн*) на разработку НИР берутся в размере 45% от основной заработной платы разработчиков НИР для покрытия административно–хозяйственных и других расходов

Кн= 127 \* 0,45 = 57,15 у.е.

3.5.7 Общая величина кап. вложений на реализацию проекта

Так как при реализации данной задачи не производилось специальных закупок техники и переустройства рабочих мест, капитальные вложения на реализацию задачи *Кр*=0 и общая величина капитальных вложений определяется затратами на предпроизводственные затраты. Общая величина капитальных вложений приведена в таблице 3.3.

Таблица 3.3 *–* Общая смета затрат на проектирование

|  |  |  |
| --- | --- | --- |
| Статьи | Затраты | |
| Сумма, (у.е.) | Удельный вес статьи в общей стоим., % |
| Материалы и покупные полуфабрикаты | 23,05 | 8,22 |
| Основная заработная плата | 127 | 45,32 |
| Дополнительная заработная плата | 12 | 4,28 |
| Отчисления на соцстрах | 55 | 19,62 |
| Затраты на отладку программы | 6 | 2,14 |
| Накладные расходы | 57,15 | 20,39 |
| ИТОГО: | 280,2 |  |

3.5.8 Расчет эксплуатационных затрат

К затратам текущего характера относятся затраты, связанные с обеспечением нормального функционирования, разработанного ПС, например, затраты на ведение информационной базы, эксплуатацию технических средств, реализацию процесса обработки информации по задачам и др.

Затраты, по эксплуатации задачи вычисляются по формуле:

(3.4)

где – *Смч*  – стоимость одного часа работы технических средств; *Тэ* – время эксплуатации задачи в течение года.

Стоимость одного часа работы технических средств возьмем равную 0,1 у.е., а время эксплуатации задачи в течение года будет равно: *Тэ*=130 ч. Подставляя фактические значения, полученные в ходе опытной эксплуатации задачи, получаем величину годовых эксплутационных расходов:

Сфз= 0,1 \* 130 = 13 у.е.

#### 3.5.9 Определение экономической эффективности внедренной задачи

Экономический эффект, как реальная экономия, обусловлена следующими факторами: сокращение недостач и порчи технических средств; сокращением потерь рабочего времени.

Рассчитаем абсолютную годовую экономию на основе сокращения потерь рабочего времени, образующуюся в виде экономии на заработной плате за счет: снижения затрат на оплату простоев служащих; сокращения численности служащих; увеличения эффективности фонда времени одного служащего; сокращения сверхурочных работ.

Сокращения затрат в сфере управления при использовании ПС решения поставленной задачи обусловлено снижением трудоемкости работ по обработке информации и снижение на оплату простоев служащих.

3.5.10 Расчет экономии за счет снижения трудоемкости решения задачи

Экономия за счет снижения трудоемкости расчета, при внедрении ПС для его решения, рассчитывается по формуле:

(3.5)

где – *А* – коэффициент, учитывающий дополнительную заработную плату; *В* – коэффициент, учитывающий отчисления на соцстрах; *Тр* – трудоемкость решения задачи вручную (ч); *Зчас* – среднечасовая тарифная ставка работника (у.е.); *Кр* – коэффициент использования технических средств; *Тоб* – трудоемкость при автоматизированной обработке (ч); *Смч*– стоимость одного машинного часа работы (у.е.); *Uе* – периодичность решения задачи (раз/год).

Подставляя фактические данные, полученные в результате исследований при ручном и автоматизированном решении задач обработки статистических данных научными работниками, получаем величину экономии за счет снижения трудоемкости решения задачи при условии:

*А* = 1,3; *В* = 2,39; *Тр* = 33; *Зчас*= 0,32 у.е. (при основной заработной плате 60 у.е., 8 часовом рабочем дне, 22 рабочих дня в месяц);

*Кр* = 1,16; *Тоб* = 0,5 ч (набор данных и расчет происходит за 30 мин.);

*Смч* = 0,1 у. е.; *Ue* = 65 раз/год.

*Этр* = (1,3 \* 2,39 \* 33 \* 0,32 – 1,16 \* 0,5 \* 0,1) \* 65 = 1329 у.е.

3.5.11 Определение годового экономического эффекта

Основной экономический показатель, определяющий экономическую целесообразность затрат на создание ПС – это экономический эффект.

##### Годовой экономический эффект определяется по формуле:

(3.6)

где – *Этр* – годовая экономия от применения внедренной задачи; *Ен*– нормативный коэффициент экономической эффективности капитальных вложений; *Кн* – единовременные затраты, связанные с внедрением задачи;

Подставляя в формулу фактические данные, определяем величину годового экономического эффекта при *Кн* = 10 у.е. без затрат, пошедших на оплату специалиста по установке:

Эс= 1329– 0,15 \* 10 – 13 = 1314,5 у.е.

3.5.12 Расчет экономической эффективности

Экономическая эффективность капитальных вложений, связанных с разработкой и внедрением ПС определяется по формуле:

Ерс = Эс / Кп (3.7)

Подставляя в формулу фактические данные, определяем величину экономической эффективности:

Ерс = 1314,5 / 388,61 = 3,38.

Так как *Ерс* ***>*** *Ен*, то внедрение экономически эффективно.

Определяем срок окупаемости внедренной задачи:

Тс = Кп / Эс = 388,61 / 1314,5 = 0,295 года

Расчеты показали, что автоматизация задачи является экономически оправданной и ведет к сокращению потерь рабочего времени за счет уменьшения времени решения «вручную», что в свою очередь приводит к значительной экономии человеческих ресурсов и финансовых средств.

**3.6 Охрана труда**

## **3.6.1 Организация рабочего места программиста**

Производственная деятельность программиста, заставляет его продолжительное время находиться в сидячем положении, которое является вынужденной позой, поэтому организм постоянно испытывает недостаток в подвижности и активной физической деятельности [16]. Работа с компьютером характеризуется значительным умственным напряжением, высокой напряженностью зрительной работы и достаточно большой нагрузкой на мышцы рук при работе с клавиатурой ЭВМ. Большое значение имеет рациональная конструкция и расположение элементов рабочего места, что важно для поддержания оптимальной рабочей позы оператора. В процессе работы с компьютером необходимо соблюдать правильный режим труда и отдыха. В противном случае отмечаются значительное напряжение зрительного аппарата с появлением жалоб на головные боли, раздражительность, нарушение сна, усталость и болезненные ощущения в глазах.

При организации рабочего места программиста должны быть соблюдены следующие основные условия: оптимальное размещение оборудования, входящего в состав рабочего места и достаточное рабочее пространство, позволяющее осуществлять все необходимые движения и перемещения.

Антропологические характеристики человека определяют габаритные и компоновочные параметры его рабочего места, а также свободные параметры отдельных элементов. По условиям работы рабочее место программиста относится к индивидуальному рабочему месту для работы сидя.

Рабочее место программиста должно занимать площадь не менее 6 м², высота помещения должна быть не менее 4 м, а объем - не менее 20 м3 на одного человека. Рабочее место программиста следует организовать следующим образом: высота над уровнем пола рабочей поверхности, за которой работает оператор, должна составлять 720 мм. Желательна возможность регулировки рабочего стола оператора по высоте в пределах 700 - 800 мм. Оптимальные размеры поверхности стола 1600 х 1000 мм2. Под столом должно иметься пространство для ног с размерами по глубине 650 мм. Рабочий стол оператора должен также иметь подставку для ног, расположенную под углом 15 к поверхности стола с размерами 400 х 350 мм. Удаленность клавиатуры от края стола должна быть не более 300 мм. Расстояние между глазами оператора и экраном видеодисплея должно составлять 40 - 80 см.

Рабочий стул программиста должен быть снабжен подъемно-поворотным механизмом. Высота сиденья должна регулироваться в пределах 400 - 500 мм. Глубина сиденья должна составлять не менее 380 мм, а ширина - не менее 400 мм. Высота опорной поверхности спинки не менее 300 мм, ширина - не менее 380 мм. Угол наклона спинки стула к плоскости сиденья должен изменяться в пределах 90 - 110.

Основным источником электромагнитных полей является монитор. Основной мерой по борьбе с электромагнитными полями является максимальное разнесение человека и монитора друг от друга. Электромагнитное поле, генерируемое монитором, вызывает электризацию пластмассовых деталей перед ним, поэтому не рекомендуется оснащать мебелью из пластмасс. Все оборудование должно быть заземлено. Допустимый уровень напряженности электростатического поля не должен превышать 20 кВ/м. В качестве меры защиты от статического электричества проводится влажная уборка помещения. Также для защиты от электромагнитных излучений используется защита временем, расстоянием и экранирование.

## **3.6.2 Расчет выносного заземления**

Рассчитаем выносное заземляющее устройство. Преимуществом такого типа заземляющего устройства является возможность выбора места размещения электродов с наименьшим сопротивлением грунта (сырое, глинистое).

Рассчитаем сопротивление группового заземлителя, если:

* мощность установки менее 2 кВА;
* вертикальный заземлитель – стальной прут ○ 25 мм и длиной 3 м;
* горизонтальный – стальная полоса шириной 25 мм, толщиной 5 мм;
* удельное сопротивление грунта (глина) 70 Ом\*м.

Рассчитаем сопротивление одиночного вертикального заземлителя:

 (3.8)



где - *ρ* - удельное сопротивление грунта (Ом\*м); *l* - длина вертикального заземлителя (м); *d* - диаметр вертикального заземлителя (м); *t* - глубина заложения.



(3.9)



Расстояние между заземлителями (м):

  (3.10)



Ориентировочное количество вертикальных заземлителей (шт):

  (3.11)

где - *Rзаз* - нормируемая вел-на сопротивления заземления (*Rзаз*=4 Ом);



Уточним количество вертикальных заземлителей (шт.):

  (3.12)

где - *ηВ* – коэффициент использования вертикальных заземлителей (так как ориентировочное *n* = 6 и *l*a = 4,5, поэтому *ηВ*=0,65).



Длина горизонтального заземлителя (м):

  (3.13)



Сопротивление горизонтального заземлителя рассчитаем по формуле:



(3.14)

где - *b1* - ширина полосы (м);



Сопротивление группового заземлителя:



(3.15)

где - *ηГ* – коэфф. использования горизонтальных заземлителей (*ηГ* =0,4).





Рассчитанное заземление подходит для проектируемого вычислительного центра и обеспечит защиту персонала от поражения электрическим током.

**3.7 Выводы**

Была рассмотрена реализация программного продукта, описаны основные алгоритмы работы, а также описан интерфейс. Проведено тестирование, выявлены и устранены основные ошибки программы. Описаны основные требования к организации рабочего места и распорядка дня.

Основными пунктами рассмотрения стали: описание реализации;

* описание пользовательского интерфейса;
* тестирование и оценка надежности программного продукта;
* охрана труда.

**ЗАКЛЮЧЕНИЕ**

В результате выполнения поставленной задачи были решены следующие частные вопросы.

Актуальность задачи – при проведении анализа предметной области, был сделал вывод что лучшим способом хранения данных в данном случае является база данных. Так как хранилище позволяет совместное использование информации многими сотрудниками и очень проста в использовании.

Существующие методики для решения данной задачи не мало – поскольку решаемая задача не является уникальной в своем роде, рассмотрены различные варианты:

* технологи штрихового кодирования;
* биометрические технологии;
* радиочастотная идентификация;
* технологии речевого ввода данных.

Было решено использовать радиочастотную технологию. Так как общедоступно, интересно и не дорого.

Выбранные следующий средства реализации:

* *Microsoft Visual Studio Enterprise* 2015*.*
* *Microsoft SQL Server.*
* *Arduino.*
* *ADO.NET.*

Перечисленные средства реализации имеют большой сборник литературы по работе с ними и доступна бесплатная подписка для студентов.

Изучили новую технологию и аппаратную часть. Из аппаратной части научились программировать микроконтроллер. Так же из технологии изучили *RFID* – *Radio Frequency Identification* (радиочастотная идентификация) с помощью которой и считывается документ, снабженный радио чипом.

Исходя из всего вышеперечисленного, была разработана программно-аппаратная часть и информационная система.

**ПЕРЕЧЕНЬ УСЛОВНЫХ ОБОЗНАЧЕНИЙ, СИМВОЛОВ,**

**ЕДИНИЦ И ТЕРМИНОВ**

БД – база данных.

ДР– дипломная работа.

ИС – информационная система.

МК – микроконтроллер.

НИР – научно-исследовательская работа.

ПО – программное обеспечение.

ПС – программные средства.

ПК – персональный компьютер.

ПИ – пользовательский интерфейс.

СУБД – система управления базой данных.

ЭВМ – электронная вычислительная машина.

*RFID* - *Radio Frequency Identification* (радиочастотная идентификация).

**СПИСОК ИСПОЛЬЗОВАННОЙ ЛИТЕРАТУРЫ**

* 1. Бейзер Б. Тестирование черного ящика. – Питер, 2005. – 320 с.
  2. БиллигВ. А. Основы программирования на *C*#. – М.: Изд-во «Интернет-университет информационных технологий – ИНТУИТ, 2006. – 488 c.
  3. ВатсонК*.* *C*#. – М.: Лори, 2004. – 880 с.
  4. ВиртН. Алгоритмы и структуры данных. – Невский диалект, 2001. – 352 с.
  5. Программирование микроконтроллерных плат *Arduino* [Электронный ресурс]. Режим доступа: *http://arduinoplus.ru/apps/programmirovanie-plat-arduino-freeduino.zip*
  6. Гуннерсон*Э.* Введение в *C*#. Библиотека программиста. – СПб.: Питер, 2001. – 304 с.
  7. КораблевВ. Самоучитель *VisualC*++ .*NET*. – СПб.: Питер; Киев: Издательская группа BHV, 2004. – 528 с.
  8. ЛибертиД. Программирование на языке *C*#. : Символ-Плюс, 2003. – 688 с.
  9. МайоД. *C*#. Искусство программирования. Энциклопедия программиста. – Киев: «ДиаСофт», 2002. – 656 с.
  10. МайоДж*.*C#. Быстрый старт. – М.: Бином, 2005. – 384 с.
  11. МикелсенК. Язык программирования *C*#. Лекции и упражнения. Учебник. – Киев: «ДиаСофт», 2002. – 656 с.
  12. *Microsoft SQL Server* 2012 руководство [Электронный ресурс]. Режим доступа *http://turbobit.net/ls71pgu3xa3h.html*
  13. ПетцольдЧ*.* Программирование для *MS Windows* на *C*#. Том 1. – М.: Издательско-торговый дом «Русская Редакция», 2002. – 576 с.
  14. Тамре Л. Введение в тестирование программного обеспечения. – М.: Издательский дом «Вильямс», 2003. – 368 с.
  15. СНиП II–2–80. Нормы проектирования. Противопожарные нормы проектирования зданий и сооружений – М.: Стройиздат, 1981 –16 с.
  16. СНиП 21–01–97. Пожарная безопасность зданий и сооружений.

**ПРИЛОЖЕНИЕ А**

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Приднестровский государственный университет им. Т.Г. Шевченко | | | | | | | | | |
| Инженерно-технический институт | | | | | | | | | |
| Инженерно-технический факультет | | | | | | | | | |
| Кафедра информационных технологий и автоматизированного управления  производственными процессами | | | | | | | | | |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  | | | |
|  |  |  |  |  |  |  | | | |
|  |  |  |  |  |  |  | | | |
|  |  |  |  |  |  |  |  |  | |
|  |  |  |  |  |  |  | | | |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
| **ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА** | | | | | | | | | |
|  | | | | | | | | | |
| **тема: «РАЗРАБОТКА ИНФОРМАЦИОННОЙ**  **СИСТЕМЫ С ПРИМЕНЕНИЕМ ТЕХНОЛОГИИ RFID»** | | | | | | | | | |
|  | | | | | | | | | |
| БАКАЛАВРСКАЯ РАБОТА | | | | | | | | | |
|  | | | | | | | | | |
|  | | | | | | | | | |
| Направление 09.03.02 «Информационные системы и технологии» | | | | | | | | | |
| Профиль: «Безопасность информационных систем» | | | | | | | | | |
|  |  |  |  |  |  |  |  |  |  |
| Руководство пользователя | | | | | | | | | |
|  | | | | | | | | | |
| Эксплуатационный документ | | | | | | | | | |
| в текстовом виде | | | | | | | | | |
| на 8 листах | | | | | | | | | |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  | Студент | |  |  |  |  | Талпа | | |
|  | группы ИТ13ДР62ИС1 | | |  |  |  | Роман Витальевич | | |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  | | |  |  |  |  | | |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
| Тирасполь, 2017 | | | | | | | | | |

**ОГЛАВЛЕНИЕ**

|  |  |  |
| --- | --- | --- |
| ВВЕДЕНИЕ | | 56 |
| 1 РУКОВОДСТВО ПО ЭКСПЛУТАЦИИ | | 57 |
|  | 1.15Назначение программы | 57 |
|  | 1.21Системные требования | 57 |
|  | 1.31Минимальный набор программного обеспечения, требуемый для работы программного продукта  1.4 Описание процедуры установки программного продукта | 57  58 |
| 2 РУКОВОДСТВО ПОЛЬЗОВАТЕЛЯ | | 59 |
|  | 2.1 Основное окно программного продукта | 59 |
|  | 2.2 Настройки | 59 |
|  | 2.3 Поиск  2.4 Работа с данными | 60  61 |
|  | |  |

**ВВЕДЕНИЕ**

Целью программно-аппаратного продукта является взаимодействие с информационной системой, для удобства и простоты использования.

Информационные системы обеспечивают сбор, хранение, обработку, поиск, выдачу информации, необходимой в процессе принятия решений задач из любой области. Они помогают анализировать проблемы и создавать новые продукты.

Так же применяется технология радиочастотного считывания. Это технология нового поколения, основанная на использовании радиочастотного электромагнитного излучения. *RFID* применяется для идентификации.

Данный программно-аппаратный продукт позволяет пользоваться бесконтактной технологией и информационной базой данных. Так как документ, снабжен чипом, который можно считать при помощи радиочастотного электромагнитного излучения. Это позволяет сэкономить время и исключить ошибки.

**1 РУКОВОДСТВО ПО ЭКСПЛУТАЦИИ**

**1.1 Назначение программы  
2**

Основой деятельностью программно-аппаратного продукта является облегченный поиск угнанных транспортных средств, проверка поддельности документов, неоплаченные штрафы и т.д. Подлинность документа легко распознать по защитной голограмме, а также при помощи информационной базы данных проверить по номеру удостоверения. Проверять каждый документ по базе данных отбирает не мало времени, что приносит дискомфорт обоим лицам, поэтому большинство документов одобряют по внешнему виду и голограмме.

**1.2 Системные требования  
2**

Для установки, разрабатываемой информационной программно-аппаратной системы необходим следующий состав технических средств:

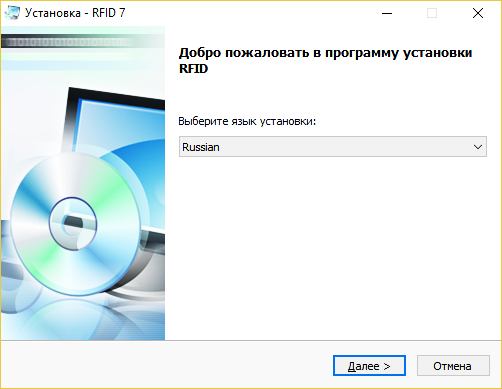
* процессор не менее *Pentium-2.0Hz;*
* оперативную память объемом не менее 512 Мб;
* устройство ввода: клавиатура, мышь, *RFID* или сенсорный ввод;
* операционная система *Windows 7, Windows 8, Windows 8.1, Windows 10*;
* компоненты среды *Net. Framework 4.0*;
* модуль *Bluetooth*.

**1.3 Минимальный набор программного обеспечения, требуемый для   
работы программного продукта  
2**

* Компоненты среды *Net.Framework 4.0*.
* Систему управления базами данных *Microsoft SQL*.
* Драйвер и модуль *Bluetooth*.

**1.4 Описание процедуры установки разработанного   
программного продукта  
2**

Для установки необходимо запустить файл *setup.exe* после чего следовать инструкциям мастера установки программы. Представлено на рисунке 1.

  
**2**  
Рисунок 1 – Окно установки

После окончания установки программы необходимо установить компоненты, требующиеся для ее работы, а именно, *Net.Framework 4.0,* модуль и драйвер *Bluetooth*, *Microsoft SQL*.

**2 РУКОВОДСТВО ПОЛЬЗОВАТЕЛЯ**

**2.1 Основное окно программного продукта  
2**

Главное окно программного продукта представлен на рисунке 2.

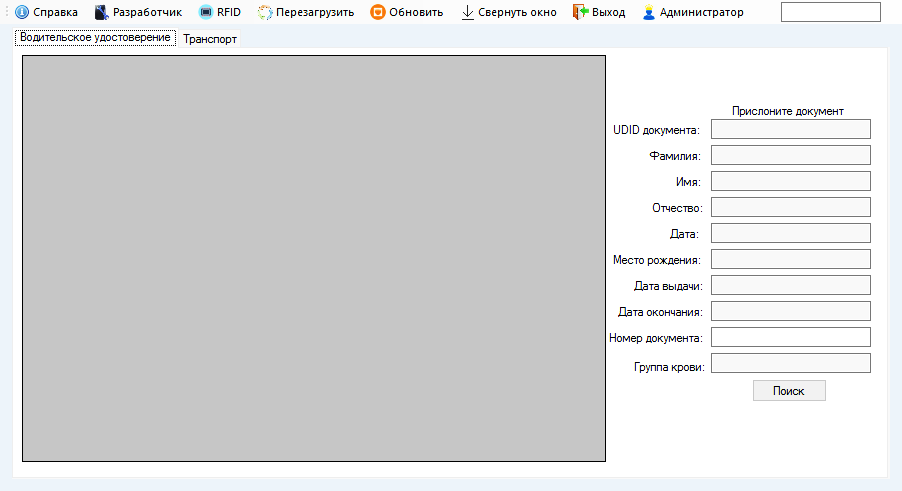
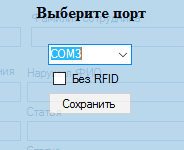


Рисунок 2 – Главное окно

На главном окне присутствуют такие элементы как «справка – информация по работе с программой», «разработчик – данные о разработчике», «*RFID* – настройки по выбору порта для работы устройства как дистанционно так и стационарно», «перезагрузить – позволяет перезапустить приложение», «обновить – позволяет обновить данные базы данных и подгрузить последние изменения», «свернуть окно – позволяет свернуть окно в трей», «выход – закрывает программу, соединение с базой данных и аппаратом», «администратор – при вводе пароля переходим в привилегированный режим который позволяет обрабатывать информацию».

**2.2 Настройки  
2**

Для работы аппаратной части требуется открытый порт, по которому будет пересылаться уникальный номер документа. Настройки представлены на рисунке 3.

  
  
Рисунок 3 – Настройки порта

Так же бывают непредвиденные ситуации и программный продукт может работать без аппаратной части, требуется поставить галочку без *RFID*.

**2.3 Поиск  
2**

Для поиска информации гражданина или транспорта достаточно две вкладки. В данных вкладах разрешено только искать информацию и не более. Рассмотреть можно на рисунке 4.

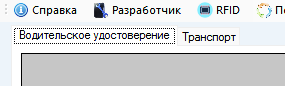


Рисунок 4 – Поиск информации

Поиск позволяет распознать документ бесконтактной технологией, что определяет подлинность документа. Так же выводится полностью информация в соответствии с документом и дополнительный список с нарушениями и статусом оплаты на рисунке 5.

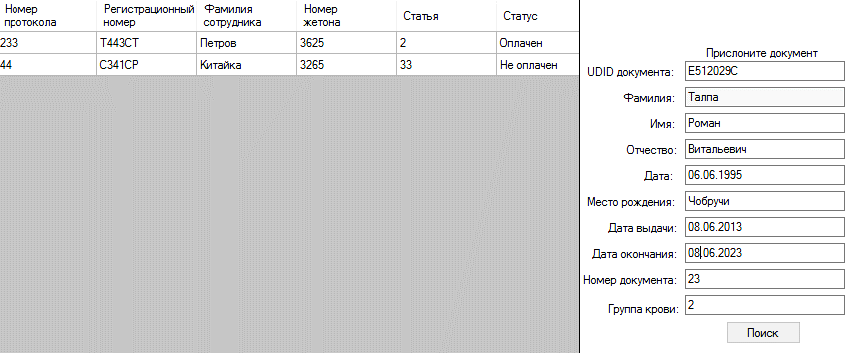


Рисунок 5 – Поиск документа

**2.4 Работа с данными  
2**

При вводе пароля отображаются дополнительные три вкладки, предназначенные для администратора, которые позволяют обрабатывать информацию. Представлено на рисунке 6.

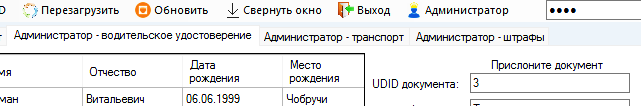


Рисунок 6 – Привилегированный режим

Для администратора поиск, редактирование, добавление и удаление представлено на рисунке 7.

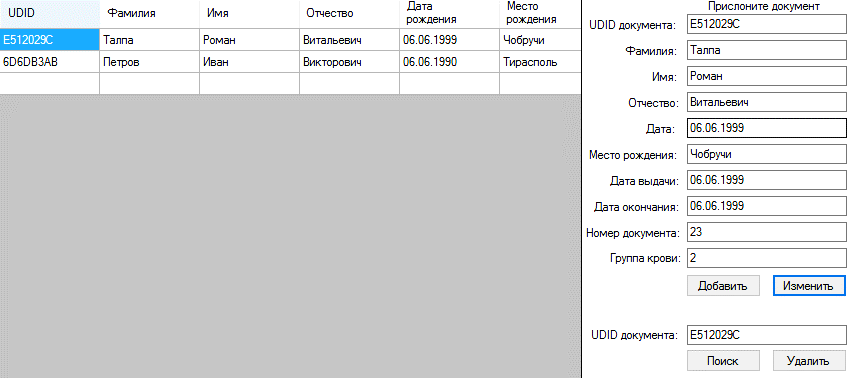


Рисунок 7 – Режим администратора

Дизайн, расположение элементов и их значение описано. Программа проста дизайном и принципом работы.