

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное автономное образовательное

учреждение высшего образования

«ЮЖНО-УРАЛЬСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

(НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ)»

Факультет электротехнический

Кафедра автоматики

Направление подготовки 27.03.04 «Управление в технических системах»

ДОПУСТИТЬ К ЗАЩИТЕ

Заведующий кафедрой

_____ С.С. Голощапов

« _____ » _____ 2018 г.

Автоматизированная система доступа в здание

(тема)

ПОЯСНИТЕЛЬНАЯ ЗАПИСКА
К ВЫПУСКНОЙ КВАЛИФИКАЦИОННОЙ РАБОТЕ

ЮУрГУ – 27.03.04.2018.281.00.00 ПЗ ВКР

Автор проекта

студент группы _____ МиЭТ-420

_____ / А.К. Гарелкин

подпись _____ ФИО

_____ 20 _____ г.

Руководитель работы

Технический директор

должность

_____ / С.Д. Клименко

подпись _____ ФИО

_____ 20 _____ г.

Нормоконтроль

Старший преподаватель

должность

_____ / В.П. Елисеев

подпись _____ ФИО

_____ 20 _____ г.

ОГЛАВЛЕНИЕ

ВВЕДЕНИЕ.....	4
ТЕОРЕТИЧЕСКИЕ ОСНОВЫ СИСТЕМ КОНТРОЛЯ УПРАВЛЕНИЯ ДОСТУПОМ.....	5
1.1 Основные понятия о СКУД.....	5
1.1.1 Определение и основные возможности систем контроля управления доступом.....	5
1.1.2 Обзор источников по системам контроля и управления доступом..	12
1.1.3 Обзор СКУД – Папилон	14
1.2 Основные понятия о средствах, используемых в работе	16
Базы данных.....	16
1.2.1 Текстовые базы данных	16
1.2.2 Сетевые базы данных	20
1.2.3 Реляционные базы данных.....	21
1.3 Выбор базы данных для использования в проектировании СКУД....	27
1.3.1 MySQL	28
1.3.1. Access	30
1.3.1. PostgreSQL	32
1.4. Обзор языков и средств, использованных в работе.....	37
1.4.1. C++.....	37

Изм.	Лист	№ докум.	Подпись	Дата

27.03.04.2018 222.00.00.ПЗ

Лист

2

1.4.2. РНР	38
1.5. Обзор контроллеров Arduino Uno	39
2.1 Составление блок схемы работы системы	44
2.2 Програмное решение чтения меток	44
2.5 Требования к информационной безопасности программной системы	48
2.3 Серверная часть	49
2.4 Интерфейс.....	51
2.4.1 Описание основных алгоритмов.....	53
Алгоритм: «блокировка пользователя по ip».....	53
Алгоритм учёта времени	54
2.5 Тестирование	56
2.6.Недостатки системы и возможные варианты улучшений	58
Итоги работы	59
Список использованных источников	59

ВВЕДЕНИЕ

В наше время у различных малых и крупных предприятий возникает потребность во внедрении автоматизированной системы контроля управления доступа (СКУД). Такие системы позволяют работодателям не только сократить затраты на охрану помещений, а так же позволяют увеличить эффективность работников на рабочем месте. Данные системы позволяют контролировать время прихода и ухода работника, что существенно повышает дисциплинированность и ответственность сотрудников. Так же системы автоматизированного контроля доступа позволяют ограничить доступ в помещения внутри предприятия посредством магнитных замков, что существенно сокращает риски краж из таких помещений как склады или хранилища, ведь доступ к материальным ценностям может быть строго ограничен и доступен только материально ответственным работникам отвечающих за эти помещения. В возможности такой системы так же может входить автоматизация въезда и выезда транспорта с территории предприятия.

Для разработки такой системы потребуется разобраться с рядом вопросов:

- Определить и рассмотреть основные понятия СКУД;
- Понять общие принципы построения;
- Рассмотреть аналогичную систему;
- Выбрать средства для разработки системы;

ТЕОРЕТИЧЕСКИЕ ОСНОВЫ СИСТЕМ КОНТРОЛЯ УПРАВЛЕНИЯ ДОСТУПОМ

1.1. Основные понятия о СКУД

Определение и основные возможности систем контроля управления доступом

Перед началом исследования существующих систем контроля и управления доступом (СКУД), нужно дать определение понятию СКУД.

Согласно ГОСТ Р 54831-2011 «Системы контроля и управления доступом. Устройства преграждающие управляемые. Общие технические требования . Методы испытаний» :[4]. СКУД – это совокупность средств контроля и управления доступом, обладающих технической , информационной, программной и эксплуатационной совместимостью.

Такая формулировка не дает нам понимания всей картины рассматриваемого понятия. Если рассмотреть более обширно СКУД - совокупность совместимых между собой аппаратных и программных средств, направленных на ограничение и регистрацию доступа людей, транспорта и других объектов в (из) помещения, здания, зоны и территории.

В большинстве случаев СКУД включает:

- устройства преграждающие управляемые (УПУ).
Например, турникеты, двери , оборудованные магнитными замками , автоматические ворота, шлагбаумы(рис. 1);



Рис.1 Турникет

- устройства считывающие, «считыватели». Например, NFC считыватели(рис. 2), дактилоскопические сканеры, сканеры радужки глаза;



Рис. 2 Считыватель

Изм.	Лист	№ докум.	Подпись	Дата

27.03.04.2018 222.00.00.ПЗ

Лист

6

- контроллеры СКУД. Электронные микропроцессорные модули, предназначенные для считывания личного идентификатора, определяющие логику авторизации для доступа в помещения с подходящим уровнем доступа;

- программное обеспечение СКУД позволяющее централизованно собирать данные и обрабатывать их.

- Узлы сети, позволяющие соединять модули системы.

- Периферия.

Одним из важных понятий для понимания работоспособности СКУД является идентификатор доступа, или идентификатор (носитель идентификационного признака): уникальный признак субъекта или объекта доступа. В качестве идентификатора может использоваться запоминаемый код, личный биометрический признак, либо вещественный код. Вещественный код – это предмет, на который с помощью специальной технологии занесен идентификационный признак в виде личного уникального кода (карты, электронные ключи) :[20].

В условиях индивидуальных особенностей экономических субъектов рассматривать возможности СКУД необходимо дифференцируя потребности каждого. Анализируя потребности банков /офисов можно выделить индивидуальные аспекты малозначимые для промышленных предприятий /заводов и государственных объектов. Тем не менее , всем СКУД присущи базовые функции, включенные в систему по умолчанию: [21]:

- ведение и поддержание баз данных пользователей , карт/идентификаторов;

- хранение фотографий пользователей в базе данных;

- фиксация даты и времени прохода в базе данных;

- задание уровней доступа;

- автономная работа контроллеров системы с сохранением основных функций управления при нарушении связи с компьютером;

- регистрация и хранение информации о событиях в энергонезависимой памяти контроллеров СКУД;
- сохранение идентификационных признаков в памяти системы при отказе и отключении электропитания;
- открывание УПУ при считывании зарегистрированного в памяти системы идентификационного признака;
- запрет открывания УПУ при считывании незарегистрированного в памяти системы идентификационного признака.

Рассмотрим дополнительные функции, которые, так или иначе, важны для всех потребителей: [14]:

1. Учет рабочего времени. С помощью программной надстройки СКУД позволяют работодателю учитывать время выхода персонала на работу и ухода с неё. Кадровые работники или специально назначенные сотрудники могут точно определить время пребывания сотрудника на рабочем месте. В начале дня встроенная в СКУД система учета рабочего времени позволяет формировать отчет о сотрудниках, не прошедших через точки доступа. Функция упрощает процедуры выявления опаздывающих или не вышедших на работу сотрудников. Это позволяет выявлять опоздавших или не явившихся на рабочее место сотрудников. Гибкость программных аналитик СКУД позволяет менеджменту формировать необходимые отчеты.

2. Возможность аппаратной и программной интеграции, регистрация и протоколирование событий в подсистемах ОПС и ССТV. Основным смыслом интеграции – это в первую очередь объединение отдельных подсистем с помощью единого интерфейса, который позволял бы выполнять и мониторинг, и настройку системы «из одного окна». Таким образом, взаимная интеграция различных систем безопасности существенно облегчает работу служб безопасности, снижает время реакции на внештатные ситуации, а также повышает эффективность ведения

расследования происшествий на объекте. Интеграция данных систем безопасности на аппаратном уровне имеет смысл в тех случаях , когда на объекте уже есть установленное оборудование и интегрировать его с новыми системами на программном уровне не представляется возможным. Кроме того, в случае оснащения небольших офисов аппаратная интеграция систем безопасности может обеспечить финансовую экономию, поскольку закупать крупную комплексную систему и ПО будет просто невыгодно.

3. Наличие SDK (Software Development Kit) для доработки функционала. Очень часто требования заказчика к комплексу систем безопасности простираются гораздо шире, чем могут предложить готовые и отработанные решения. В таких случаях приходится интегрировать большое количество разрозненных подсистем , и здесь, конечно же, не обойтись без SDK. В качестве примера систем, для интеграции которых необходим SDK, можно привести ERP-систему предприятия , систему выдачи ключей от помещений, а также ИТ- инфраструктуру организации, которая позволяет объединить логический и физический доступ.

4. Защита технических средств и ПО от НСД. Защита от несанкционированного доступа к настройкам оборудования и ПО СКУД является необходимым условием надежной и эффективной работы системы , не позволяет допускать нарушения условий допуска и работы персонала.

5. Ручное, полуавтоматическое или автоматическое открывание УПУ при ЧС. Данная функция является необходимым условием эксплуатации объектов и помещений по технике безопасности.

6. Максимальное количество идентификаторов для точки прохода . Понимается количество карточек, которые могут быть записаны на одном контроллере. Наиболее узким местом являются контроллеры турникетов на центральной проходной , так как через нее ходят все сотрудники

и посетители. Для обычного офисного здания это ограничение не играет большой роли (число сотрудников и посетителей редко превышает несколько тысяч). Однако если речь идет о предприятии с несколькими корпусами и большим количеством сотрудников, этот параметр необходимо учитывать.

7. Максимальное количество контроллеров в сети. Фактически означает максимальное количество контролируемых точек прохода в системе.

1.1.2 Общие принципы работы

Существующие системы контроля управления доступа (СКУД), условно, можно разделить на две категории:

1. Простые, рассчитанные всего на одну входную дверь. Обычно при такой архитектуре используются автономные контроллеры
2. Сложные, предназначенные для контроля доступа на крупных объектах - предприятиях, заводах и банках :[19].

Независимо от конструкции СКУД, каждая подобная система состоит из нескольких обязательных объектов, это - контроллеры для управления, считыватели для идентификации, а также всевозможные исполнительные устройства ограничения доступа: турникеты, электромагнитные замки и защелки. Электронные бесконтактные карты или ключи в качестве пропусков являются самым распространенным и удобным, а так же самым дешёвым средством осуществления в системах контроля доступа.

Принцип работы всех систем сводиться к трем базовым действиям:

1. Предоставление права доступа.
2. Определение личности и уровня доступа.
3. Доступ.

Работает система контроля и управления доступом следующим образом: после установки системы контроля доступа на объекте , каждый сотрудник получает уникальный идентификатор (магнитная карточка, бесконтактная карта, отпечаток пальца, в случае использования биометрических систем) и выставляются права доступа в различные зоны. На входе в зоны, требующие контроля , устанавливаются считыватели идентификаторов, контроллеры и электромагнитные замки.

Идентификатор может одновременно является пропуском на территорию организации и ключом от помещений , куда сотруднику разрешен доступ.

Все ограничивающие устройства подключаются к контроллерам системы управления доступом . Контроллеры предназначены для приема и анализа информации о предъявляемых картах доступа, а также для управления различными исполнительными устройствами . В состав оборудования системы контроля доступа могут входить два типа контроллеров: контроллеры замка и контроллеры турникета, каждый из которых отвечает за контроль работы собственного узла.

Для прохода через турникет или входа в ответственное помещение работники предприятия должны поднести свою карту доступа к считывателю, после чего считыватель передает код предъявленной карты в контроллер , а контроллер доступа принимает решение о разрешении или запрете прохода на основании заложенной в него информации. В случае если доступ разрешен, система контроля доступа автоматически разблокирует турникет или замок на двери.

Также с помощью СКУД можно осуществлять контроль въезда автотранспорта на территорию объекта, в этом случае после предъявления персонального идентификатора происходит открытие ворот или подъем шлагбаума.

Все события о проходах через контрольные пункты фиксируются в памяти системы управления доступом и могут использоваться

					27.03.04.2018 222.00.00.ПЗ	Лист
Изм.	Лист	№ докум.	Подпись	Дата		11

для автоматизированного учета рабочего времени , блокировок учетных записей во внутренней компьютерной сети, а также для получения отчетов по дисциплине труда или для возможных служебных расследований на предприятии.

1.1.2 Обзор источников по системам контроля и управления доступом

В книге « Системы контроля и управления доступом» [3] изложен широкий круг вопросов, связанных с организацией контрольно-пропускного режима на различных объектах и применением СКУД. Большое внимание уделено средствам идентификации и аутентификации. Описаны устройства идентификации (считывания) различных типов ; средства биометрической аутентификации личности и особенности их реализации; различные виды контроллеров и исполнительные устройства СКУД. Приведен обзор различных вариантов реализации СКУД . Даны основные рекомендации по выбору средств и систем контроля доступа. В приложении приведены ключевые выдержки из официальных нормативных материалов связанных с использованием СКУД.

В книге « Системы безопасности и устройства кодового доступа. Просто о сложном» [10] приведены описания стандартов в спецификациях Международного комитета по стандартизации ISO 18902, ISO 7816, ISO 14443 варианты А и В, ISO/IEC 15693, 180 15693-2, спецификации ЕМУ (Europay, MasterCard, Visa), IPC/JEDEC J-STD-020C, ECMA 340, ETSI TS 102190 и др.

Системы, работающие по технологии Java, регламентируются и описываются в стандартах Java Card 2.1.1, Java Card 2.1.1 и выше. Рассмотрены технические характеристики наиболее популярных микроконтроллеров , которые могут пригодиться разработчикам и пользователям систем безопасности и кодового доступа. Собраны актуальные сведения по новейшим системам доступа

и безопасности, конфигурации систем СКУД, рассмотрены особенности «меток» различных стандартов, их технические характеристики и справочные данные по микроконтроллерам смарт-карт. В третьей главе книги представлены новые радиолобительские схемы и избранные электронные устройства-помощники по теме охраны и контроля доступа.

В книге « Электронная идентификация» [6] рассматриваются актуальные вопросы создания и применения электронных идентификационных средств для обеспечения достоверности учета и контроля во всех сферах жизнедеятельности человека . Подробно анализируются технологии бесконтактной радиочастотной идентификации с индуктивной, электромагнитной и емкостной связью между мобильными носителями электронных данных и считывающими устройствами. Рассматриваются перспективные системы электронной идентификации, основанные на использовании эффекта поверхностных акустических волн. Значительное внимание уделяется таким перспективным средствам электронной идентификации, как интеллектуальные смарт-карты. Описываются принципы, современные алгоритмы и протоколы криптографической защиты электронных идентификаторов, в частности смарт-карт. Приводятся многочисленные примеры практического применения средств электронной идентификации в разнообразных областях жизнедеятельности человека.

Учебное пособие « Системы контроля и управления доступом» [2] рассматривает безопасность и защиту информационных ресурсов, материальных ценностей и коммерческой тайны, дисциплину сотрудников как совокупность необходимых условий нормального существования и успешного развития любого предприятия. Авторами рассматриваются системы контроля и управления доступом. Анализируются особенности функционирования , основные характеристики и параметры, которые целесообразно учитывать при проектировании системы,

выборе алгоритма ее работы и конкретной аппаратуры для реализации.

Учебное пособие « Проектирование и исследование комплексных систем безопасности» :[17] рассматривает все компоненты обеспечения безопасности. Автор детально описывает каждую составляющую. Проводит обзор, в том числе устройств идентификации , видов исполнительных устройства и контроллеров СКУД . Дается оценка влияния СКУД на обеспечение безопасности объекта.

1.1.3 Обзор СКУД – Папилон

Общие положения:

PAPILLON GWI - это система, предназначенная для реализации полного комплекса мероприятий по организации пропускной системы внутри компании и учета заработной платы в компании.

Система делает возможным решение ряда задач:

- контроль доступа (входа и выхода) на территорию компании и подавление несанкционированного доступа;
- контроль движения персонала и транспортных средств по территории компании;
- контроль и ограничение доступа сотрудников внутри предприятия к помещениям со статусом повышенной безопасности или секретности в соответствии с уровнем доступа каждого сотрудника;
- хранить записи о рабочем времени сотрудников и нарушениях плана работы.

Система предусматривает создание электронной базы данных (БД) сотрудников компании с их распределением по отделам, а также лицами, имеющими право на различный уровень доступа в различные помещения или отделы компании.

В дополнение к подробной учетной информации для каждого сотрудника (фото, ФИО, должность , номер отдела, уровень доступа,

					27.03.04.2018 222.00.00.ПЗ	Лист
Изм.	Лист	№ докум.	Подпись	Дата		14

рабочий план), его личный уникальный идентификатор зарегистрирован в базе данных.

Уникальный идентификатор сотрудника в системе GWI PAPILLON является уникальным идентификационным кодом, выданным сотруднику, когда его регистрировали в базе данных и закодирован на личной карте сотрудника.

Необходимым условием для функционирования системы Папилон GWI является организация ввода данных с различных датчиков по всему периметру компании и на границе помещений (зон территорий), которые регулируются внутренними уставом и уровнем доступа сотрудников.

Контрольно-пропускные пункты оснащены специальными устройствами для считывания личных идентификаторов - бесконтактными считывателями данных с карты (считыватели производятся внутри компании).

Система поддерживает работу контрольных точек без ограничений физического прохода и контрольных точек , оборудованных контролируемые запираемыми устройствами магнитными замками , шлагбаумами, автоматическими дверями.

Точки без физических ограничений транзита в основном используются для сбора информации о движениях сотрудников внутри предприятия и позволяют отслеживать местоположение сотрудников в режиме реального времени, а также фактическое рабочее время работников, а так же расходы внутри компании (столовые, спортивные залы, автомойки и т.д.). Факт пересечения контрольной точки определяется считателем, информация передается на локальный компьютерный мост для определения и далее в пункт безопасности и системный сервер для записи события в БД (идентификация и время) контрольная точка для обработки и отчетности. Все работники предприятия обязаны постоянно пользоваться системой, данная обязанность определяется внутренним уставом компании.

					27.03.04.2018 222.00.00.ПЗ	Лист
Изм.	Лист	№ докум.	Подпись	Дата		15

При попытке пересечь контрольно-пропускной пункт , который оснащен точкой, информация передается от считывателя к серверу системы или локального компьютера шлюза, где идентифицируется работник и в соответствии со своими уровнем доступа, а также графиком доступности помещения, решение принимается системой самостоятельно, чтобы разрешить или запретить проход. В зависимости от системы влияние на привод запирающего устройства происходит автоматически или по команде с пункта охраны. Вся информация находится на сервере системы , как факты успешного пересечения такой контрольной точки проведены (данные о работнике), а также факты неудачных попыток времени пересечения.

1.2. Основные понятия о средствах, используемых в работе

Базы данных

1.2.1 Текстовые базы данных

Объектами хранения в текстовых БД являются тексты. Под текстом будут пониматься неструктурированные данные, построенные из строк.

Основной целью любой текстовой БД является хранение, поиск и выдача документов, соответствующих запросу пользователя. Такие документы принято называть релевантными. Ввиду того, что автоматизированный поиск документов на естественных языках достаточно затруднен, возникает вопрос о проектировании некоторых формальных языков , предназначенных для отображения основного смыслового содержания документов и запросов в БД.

Такие языки называют информационно-поисковыми. В настоящее время разработано достаточно большое количество информационно-поисковых языков, которые отличаются не только по своим изобразительным свойствам, но и по степени семантической силы.

					27.03.04.2018 222.00.00.ПЗ	Лист
Изм.	Лист	№ докум.	Подпись	Дата		16

В основе подхода к построению классификационных языков лежит представление о том, что накопленные знания могут быть разделены на взаимоисключающие классы и подклассы. Существует система правил, которой должен подчиняться любой язык классификационного типа, в частности:

- деление отраслей знаний на классы и подклассы проводится по одному основанию;
- подклассы должны исключать друг друга;
- при делении классов на подклассы должна соблюдаться непрерывность.

Информационно – поисковые языки, получившие название дескрипторных, основаны на применении принципов координатного индексирования, при котором смысловое содержание документа может быть с определенной степенью точности и полноты задано списком ключевых слов, содержащихся в тексте.

Дескрипторные языки привязаны к лексике текстов. Ключевые слова из текстов выбираются исходя из разных целей, соответственно, критерии выбора могут различаться. Для построения дескрипторного языка критерием отбора ключевых слов, как правило, служат информативность слова и частота его встречаемости в тексте.

Универсальными структурами дескрипторного языка являются лексические единицы, парадигматические и синтагматические отношения.

Лексическая единица – наименьшая смысловая единица, задаваемая при построении языка.

В большинстве автоматизированных информационных систем при индексировании документов и запросов применяется контроль с помощью тезауруса. Контроль может осуществляться в автоматизированном или ручном режиме. По сути дела тезаурус представляет собой словарь – справочник, в

котором присутствуют все лексические единицы дескрипторного информационно поискового языка с введенными парадигматическими отношениями. Парадигматические отношения могут задаваться как:

- отношения вид – род (вышестоящий дескриптор);
- отношения род – вид (нижестоящие дескрипторы);
- синонимы;
- ассоциативные связи.

В тезаурусы помещаются дескрипторы и недескрипторы, хотя существуют тезаурусы только из дескрипторов.

Как дескрипторы, так и недескрипторы приводят к единой грамматической форме. Как правило, дескрипторы употребляются в форме существительных или именных словосочетаний. Тезаурус может быть построен по принципу дескрипторных статей, состоявших из заглавного дескриптора и списка дескрипторов и недескрипторов с обозначением парадигматических отношений. Тезаурус может быть двуязычным. В этом случае эквивалентный дескриптор на иностранном языке должен быть обозначен.

Парадигматические отношения представляют собой внетекстовые отношения между лексическими единицами. На их основании происходит группировка лексических единиц в парадигмы.

Синтагматические отношения представляют собой отношения лексических единиц в тексте, т.е. они выражают семантику контекста.

При переводе основного смыслового содержания документов и запросов с естественного языка на дескрипторный информационно – поисковый язык существуют определенные правила, называемые системой индексирования. Результатом перевода документа является поисковый образ документа, а запроса – поисковый образ запроса.

Из перечисленных информационно – поисковых языков именно дескрипторные языки наилучшим образом приспособлены для описания документов и запросов при автоматизированном поиске в текстовых БД. Языки эти обладают таким преимуществом, как гибкость, открытость, близость к естественному языку ; это языки двухуровневые (уровень ключевых слов и уровень дескрипторов). Дескрипторные информационно – поисковые языки позволяют формулировать документы и запросы в разных терминах. К основным недостаткам языков данного класса можно отнести недостаточную полноту описания смыслового содержания документов и запросов.

Системы, контролируемые тезаурусом , содержат процедуры как морфологического, так и синтаксического анализа текстов. Однако при проектировании ряда БД возникает необходимость в добавлении еще одного этапа анализа текста на естественном языке – анализа его семантической структуры. Примером таких баз могут быть БД, ориентированные на поиск по образцам. В подобных семантических системах пытаются моделировать процесс понимания законченных описаний фрагментов действительности , например патентов, рассказов, эпизодов и др., выраженных в виде текстов. Как правило, понимание текста трактуется как процесс извлечения из него существенной с точки зрения системы информации. Извлеченная информация вводится в базу знаний , представляющую собой динамическую информационную модель реального мира. Затем система способна отвечать на запросы относительно событий, фактов, явлений, изложенных в текстах.

Пакеты прикладных программ, предназначенные для ввода , обработки, поиска и обновления текстов, называют информационно-поисковой системой (ИПС).

1.2.2 Сетевые базы данных

Одним из наиболее эффективных методов представления знаний являются сетевые модели.

В основе моделей лежит понятие сети, вершинами которой являются понятия, соответствующие объектам, событиям, процессам, явлениям, а дугами – отношения между этими понятиями.

Узлы и связи можно наглядно изображать в виде диаграмм.

Если вершины сети не имеют своей внутренней структуры, то сеть будет простой. Если же вершины обладают некоторой структурой в виде сети, то сеть называется иерархической. Если отношения между вершинами одинаковые, то сеть однородна, в противном случае – сеть неоднородна. Характер отношений, приписываемый дугам, может быть различен. В соответствии с этим выделяют следующие типы сетей:

- функциональные сети отражают декомпозицию определенной вычислительной или информационной процедуры, а дуги показывают функциональную связь между декомпонированными частями; этот язык недостаточно богат для представления знаний;
- сценарии, представляющие собой однородные сети с единственным отношением в виде нестрогого порядка. Семантика отношений может быть различной. Отношение может трактоваться как классифицирующее, временное и т.п. Сценарии часто используются при формировании допустимых планов по достижению цели;
- семантические сети используют отношения разных типов, а вершины в них могут иметь разную интерпретацию. По сути дела семантическая сеть является классом, в который включаются как сценарии, так и функциональные сети. Наиболее часто используются в сети связи типа «это есть». Они позволяют построить в виде сети иерархию понятий, в которых узлы низших уровней

наследуют свойства узлов более высоких уровней. Именно таким механизмом переноса свойств обусловлена эффективность семантических сетей.

1.2.3 Реляционные базы данных

Базы данных называются реляционными, если управление ими основано на математической модели, использующей методы реляционной алгебры и реляционного исчисления. С. Дейт дает следующее неформальное определение реляционных баз данных:

- Вся информация в базе данных представлена в виде таблиц.
- Поддерживаются три реляционных оператора – выбора, проектирования и объединения, с помощью которых можно получить любые необходимые данные, заложенные в таблицы.

Доктор И.Ф. Кодд, автор реляционной модели, разработал целый список критериев, которым должна удовлетворять реляционная модель. Описание этого списка, часто называемого «12 правилами Кодда», требует введения сложной терминологии и выходит за рамки дипломной работы. Тем не менее можно назвать некоторые правила Кодда для реляционных систем. Чтобы считаться реляционной по Кодду, система управления базами данных должна:

- представлять всю информацию в виде таблиц;
- поддерживать логическую структуру данных, независимо от их физического представления;
- использовать язык высокого уровня для структурирования, выполнения запросов и изменения информации в базах данных;
- поддерживать основные реляционные операции (выбор, проектирование и объединение), а также теоретико-множественные операции, такие как объединение, пересечение и дополнение;

- поддерживать виртуальные таблицы , обеспечивая пользователям альтернативный способ просмотра данных в таблицах;
- различать в таблицах неизвестные значения (nulls), нулевые значения и пропуски в данных;
- обеспечивать механизмы для поддержки целостности, авторизации , транзакций и восстановления данных.

Первое правило Кодда гласит, что вся информация в реляционных базах данных представляется значениями в таблицах. В реляционных системах таблицы состоят из горизонтальных строк и вертикальных столбцов. Все данные представляются в табличном формате – другого способа просмотреть информацию в базе данных не существует. Набор связанных таблиц образует базу данных . Таблицы в реляционной базе разделены, но полностью равноправны. Между ними не существует никакой иерархии.

Каждая таблица состоит из строк и столбцов. Каждая строка описывает отдельный объект или сущность – ученика, предмет , день недели или что-нибудь другое. Каждый столбец описывает одну характеристику объекта – имя или фамилию ученика, его адрес, оценку, дату. Каждый элемент данных, или значение, определяется пересечением строки и столбца. Чтобы найти требуемый элемент данных, необходимо знать имя содержащей его таблицы, столбец и значение его первичного ключа, или уникального идентификатора.

В реляционной базе данных существует два типа таблиц – пользовательские таблицы и системные таблицы. Пользовательские таблицы содержат информацию, для поддержки которой собственно и создавались реляционные базы данных. Системные таблицы обычно поддерживаются самой СУБД, однако доступ к ним можно получить так же, как и к любым другим таблицам. Возможность получения доступа к системным таблицам, по аналогии с любыми другими таблицами , составляет основу другого правила Кодда для реляционных систем.

Реляционная модель обеспечивает независимость данных на двух уровнях – физическом и логическом. Физическая независимость данных означает с точки зрения пользователя, что представление данных абсолютно не зависит от способа их физического хранения. Как следствие этого, физическое перемещение данных никоим образом не может повлиять на логическую структуру базы данных. Другой тип независимости, обеспечиваемый реляционными системами - логическая независимость – означает, что изменение взаимосвязей между таблицами и строками не влияет на правильное функционирование программных приложений и текущих запросов.

В определении системы управления реляционными базами данных упоминаются три операции по выборке данных – проектирование, выбор и объединение, которые позволяют строго указать системе, какие данные необходимо показать. Операция проектирования выбирает столбцы, операция выбора – строки, а операция объединения собирает вместе данные из связанных таблиц.

Виртуальные таблицы можно рассматривать как некоторую перемещаемую по таблицам рамку, через которую можно увидеть только необходимую часть информации. Виртуальные таблицы можно получить из одной или нескольких таблиц базы данных (включая и другие виртуальные таблицы), используя любые операции выбора, проектирования и объединения. Виртуальные таблицы, в отличие от «настоящих», или базовых таблиц, физически не хранятся в базе данных. В то же время необходимо осознавать, что виртуальные таблицы это не копия некоторых данных, помещаемая в другую таблицу. Когда вы изменяете данные в виртуальной таблице, то тем самым изменяете данные в базовых таблицах. В идеальной реляционной системе с виртуальными таблицами можно оперировать как и с любыми другими таблицами. В реальном мире на виртуальные таблицы накладываются определенные ограничения, в частности на обновление. Одно из правил Кодда гласит, что в истинно реляционной системе над виртуальными таблицами можно выполнять все «теоретически» возможные операции. Большинство

современных систем управления реляционными базами данных не удовлетворяют этому правилу полностью.

В реальном мире управления информацией данные часто являются неизвестными или неполными: неизвестен телефонный номер, не захотели указать возраст. Такие пропуски информации создают «дыры» в таблицах. Проблема, конечно, состоит не в простой неприглядности подобных дыр. Опасность состоит в том, что из-за них база данных может стать противоречивой. Чтобы сохранить целостность данных в реляционной модели, так же, как и в правилах Кодда, для обработки пропущенной информации используется понятие нуля.

«Ноль» не означает пустое поле или обычный математический ноль. Он отображает тот факт, что значение неизвестно, недоступно или неприменимо. Существенно, что использование нулей инициирует переход с двухзначной логики (да/нет) на трехзначную (да/нет/может быть). С точки зрения другого эксперта по реляционным системам, Дейта, нули не являются полноценным решением проблемы пропусков информации. Тем не менее они являются составной частью большинства официальных стандартов различных реляционных СУБД.

Целостность – очень сложный и серьезный вопрос при управлении реляционными базами данных. Несогласованность между данными может возникать по целому ряду причин. Несогласованность или противоречивость данных может возникать вследствие сбоя системы – проблемы с аппаратным обеспечением, ошибки в программном обеспечении или логической ошибки в приложениях. Реляционные системы управления базами данных защищают данные от такого типа несогласованности, гарантируя, что команда либо будет исполнена до конца, либо будет полностью отменена. Этот процесс обычно называют управлением транзакциями.

Другой тип целостности, называемый объектной целостностью, связан с корректным проектированием базы данных. Объектная целостность требует, чтобы ни один первичный ключ не имел нулевого значения.

Третий тип целостности, называемой ссылочной целостностью, означает непротиворечивость между частями информации, повторяющимися в разных таблицах. Например, если вы изменяете неправильно введенный номер карточки страхового полиса в одной таблице, другие таблицы, содержащие эту же информацию, продолжают ссылаться на старый номер, поэтому необходимо обновить и эти таблицы. Чрезвычайно важно, чтобы при изменении информации в одном месте, она соответственно изменялась и во всех других местах. Кроме того, по определению Кодда, ограничения на целостность должны:

- определяться на языке высокого уровня, используемом системой для всех других целей;
- храниться в словаре данных, а не в программных приложениях.

Эти возможности в том или ином виде реализованы в большинстве систем.

1.3.4 Объектно-ориентированные базы данных

Термин "объект" в программной индустрии впервые был введен в языке Simula (1967 г.) и означал какой-либо аспект моделируемой реальности. Сейчас под объектом понимается "нечто, имеющее четко определенные границы" (определение известного американского специалиста Г.Буча). Объекты, обладающие одинаковыми свойствами, составляют классы (например, курица, пингвин и чайка - объекты класса "птицы"). Обычно класс описывается как новый тип данных, а объекты (экземпляры класса) - определенные на его основе переменных.

Сразу же необходимо заметить, что общепринятого определения "объектно- ориентированной модели данных" не существует. Сейчас можно говорить лишь о некоем "объектном" подходе к логическому представлению данных и о различных объектно-ориентированных способах его реализации.

В объектно- ориентированных базах данных, в отличие от реляционных, хранятся не записи , а объекты. ОО-подход представляет более совершенные средства для отображения реального мира, чем реляционная модель:

- естественное представление данных . В реляционной модели все отношения принадлежат одному уровню, именно это осложняет преобразование иерархических связей модели "сущность-связь" в реляционную. ОО-модель можно рассматривать послойно, на разных уровнях абстракции;

- имеется возможность определения новых типов данных и операций с ними.

В то же время, ОО-модели присущ и ряд недостатков:

- отсутствуют мощные непроцедурные средства извлечения объектов из базы. Все запросы приходится писать на процедурных языках, проблема их оптимизации возлагается на программиста;

- вместо чисто декларативных ограничений целостности (типа явного объявления первичных и внешних ключей реляционных таблиц с помощью ключевых слов PRIMARY KEY и REFERENCES) или полудекларативных триггеров для обеспечения внутренней целостности приходится писать процедурный код.

Очевидно, что оба эти недостатка связаны с отсутствием развитых средств манипулирования данными. Эта задача решается двумя способами - расширение ОО-языков в сторону управления данными (стандарт ODMG), либо способ объединения возможностей реляционного и объектно- ориентированного подхода к управлению данными. Его предложил известный американский ученый Майкл Стоунбрейкер. Согласно его воззрениям реляционную СУБД нужно просто дополнить средствами доступа к сложным данным. При этом ядро

СУБД не требует переработки, как в случае с SQL3, и сохраняет все присущие реляционным системам достоинства. Объектные расширения реализуются в виде надстроек, которые динамически подключаются к ядру. На основе этой идеи под руководством М.Стоунбрейкера в университете Беркли (Калифорния, США) была разработана СУБД Postgres.

Появление объектно-ориентированных СУБД вызвано потребностями программистов на ОО-языках, которым были необходимы средства для хранения объектов, не помещавшихся в оперативной памяти компьютера. Также важна была задача сохранения состояния объектов между повторными запусками прикладной программы. Поэтому, большинство ООСУБД представляют собой библиотеку, процедуры управления данными которой включаются в прикладную программу. Примеры реализации ООСУБД как выделенного сервера базы данных крайне редки.

Выбор базы данных для использования в проектировании СКУД Главным

аргументом в выборе СУБД является её доступность и простота использования. Под простотой использования нужно понимать, что нам нужна система с реляционной моделью, которая обладает рядом важных преимуществ:

- Независимостью логической структуры от физического представления
- Гибкостью структуры БД – конструктивные решения не ограничивают разработчиков в выполнении в будущем самых разнообразных запросов.

Реляционная модель не требует описания всех возможных связей между данными, таким образом разработчик может давать различные запросы на взаимосвязь между данными, да же на те связи которые не были заложены первоначально.

Для этого на потребуется рассмотреть различные СУБД присутствующие на рынке.

1.3.1.MySQL

MySQL является быстрой и не очень большой реляционной СУБД основанной на традициях Hughes Technologies Mini SQL (mSQL).

Программное обеспечение MySQL - это ПО с открытым кодом . ПО с открытым кодом даёт любому желающему модифицировать и изменять код. Данное ПО можно получить совершенно бесплатно . При этом каждый юзер может изучить исходный код и доработать его в соответствии со своими потребностями. Использование ПО MySQL регламентируется лицензией GPL (GNU General Public License), в которой указано, что можно делать с их ПО а чего нельзя в различных ситуациях.

MySQL является очень быстрым, надежным и легким в использовании. Это именно те качества которые нам требуются. MySQL обладает также большим функционалом, разработанных в тесном контакте с пользователями.

Первоначально сервер MySQL разрабатывался для управления большими базами данных с целью увеличить скорость работы по сравнению с аналогами. И вот уже в течение нескольких лет данный сервер успешно используется в различных структурах с высокими технологическими требованиями. Несмотря на то , что MySQL постоянно улучшается , он уже сегодня обеспечивает широкий спектр полезных функций. Благодаря своей скорости, доступности и уровню безопасности MySQL очень хорошо подходит для доступа к базам данных через Интернет.

Программное обеспечение MySQL является системой клиент- сервер , которая содержит в себе многопоточный SQL- сервер , реализующий поддержку различных вычислительных машин баз данных,

а также несколько различных клиентских программ и библиотек, средства администрирования и широкий спектр программных интерфейсов (API).

Также сервер MySQL может быть поставлен в виде многопоточной библиотеки, которую можно подключить к пользовательскому приложению и получить компактный, более быстрый и легкий в управлении продукт.

Перечислю основные приятные стороны пакета MySQL:

- многопоточность - поддержка нескольких одновременных запросов;
- оптимизация связей с присоединением многих данных за один проход;
- записи фиксированной и переменной длины;
- ODBC драйвер в комплекте с исходником;
- гибкая система привилегий и паролей;
- до 16 ключей в таблице. Каждый ключ может иметь до 15 полей;
- поддержка ключевых полей и специальных полей в операторе CREATE;
- поддержка чисел длиной от 1 до 4 байт (ints, float, double, fixed), строк переменной длины и меток времени;
- интерфейс с языками C и perl;
- основанная на потоках, быстрая система памяти;
- утилита проверки и ремонта таблицы (isamchk);
- все данные хранятся в формате ISO8859_1;
- все операции работы со строками не обращают внимания на регистр символов в обрабатываемых строках;
- псевдонимы применимы как к таблицам, так и к отдельным колонкам в таблице;
- все поля имеют значение по умолчанию;
- легкость управления таблицей , включая добавление и удаление ключей и полей.

1.3.1. Access

СУБД Access является системой управления базами данных реляционного типа. Достоинством её является то что она имеет очень простой графический интерфейс, который позволяет не только создавать собственную базу данных, но и разрабатывать простые и сложные приложения.

В отличие от других настольных СУБД, Access хранит все данные в одном файле, хотя и распределяет их по разным таблицам. Строки (записи) которых состоят из наборов полей определенных типов. Возможность создания таблиц не ограничена.

С каждой таблицей могут быть связаны индексы (ключи), задающие нужные пользователю порядки на множестве строк. Таблицы могут иметь однотипные поля (столбцы), и это позволяет устанавливать между ними связи, выполнять операции реляционной алгебры. Типичными операциями над базами данных являются определение, создание и удаление таблиц, модификация определений (структур, схем) существующих таблиц, поиск данных в таблицах по определенным критериям (выполнение запросов), создание отчетов о содержимом базы данных.

СУБД позволяет задавать типы данных и способы их хранения. Можно также задать критерии (условия), которые СУБД будет в дальнейшем использовать для обеспечения правильности ввода данных. В самом простом случае условие на значение должно гарантировать, что не будет введен случайно в числовое поле буквенный символ. Другие условия могут определять область или диапазоны допустимых значений вводимых данных.

Microsoft Access предоставляет максимальную свободу в задании типа данных (текст, числовые данные, даты, время, денежные значения, рисунки, звук, электронные таблицы). Имеется возможность задать форматы

хранения предоставления этих данных при выводе на экран или печать. Для уверенности, что в базе хранятся только корректные значения, можно задать условия на значения различной степени сложности.

Так как Microsoft Access является современным приложением Windows, можно использовать в работе все возможности DDE (динамический обмен данными) и OLE (связь и внедрение объектов). DDE позволяет осуществлять обмен данными между Access и любым другим поддерживающим DDE приложением Windows. В Microsoft Access можно при помощи макросов или Access Basic осуществлять динамический обмен данными с другими приложениями.

OLE является более изощренным средством Windows, которое позволяет установить связь с объектами другого приложения или внедрить какие-либо объекты в базу данных Access. Такими объектами могут быть картинки, диаграммы, электронные таблицы или документы из других поддерживающих OLE приложений Windows.

В Microsoft Access для обработки данных базовых таблиц используется мощный язык SQL (структурированный язык запросов). Используя SQL можно выделить из одной или нескольких таблиц необходимую для решения конкретной задачи информацию. Access значительно упрощает задачу обработки данных. Совсем не обязательно знать язык SQL. При любой обработке данных из нескольких таблиц Access использует однажды заданные связи между таблицами.

В Microsoft Access имеется также простое и в то же время богатое возможностями средство графического задания запроса – так называемый «запрос по образцу» (query by example), которое используется для задания данных, необходимых для решения некоторой задачи. Используя для выделения и перемещения элементов на экране стандартные приемы работы с мышью в Windows и несколько клавиш на клавиатуре, можно буквально за секунды построить довольно сложный запрос.

Microsoft Access спроектирован таким образом , что он может быть использован как в качестве самостоятельной СУБД на отдельной рабочей станции, так и в сети – в режиме «клиент-сервер». Поскольку в Microsoft Access к данным могут иметь доступ одновременно несколько пользователей, в нем предусмотрены надежные средства защиты и обеспечения целостности данных. Можно заранее указать , какие пользователи или группы пользователей могут иметь доступ к объектам (таблицам, формам , запросам) базы данных . Microsoft Access автоматически обеспечивает защиту данных от одновременной их корректировки разными пользователями.

Практически все существующие СУБД имеют средства разработки приложений, которые могут быть использованы программистами или квалифицированными пользователями при создании процедур для автоматизации управления и обработки данных.

Microsoft Access предоставляет дополнительные средства разработки приложений, которые могут работать не только с собственными форматами данных, но и с форматами других наиболее распространенных СУБД. Возможно, наиболее сильной стороной Access является его способность обрабатывать данные электронных таблиц, текстовых файлов, файлов dBASE, Paradox, Vtrieve, FoxPro и любой другой базы данных SQL, поддерживающей стандарт ODBE. Это означает, что можно использовать Access для создания такого приложения Windows, которое может обрабатывать данные, поступающие с сетевого сервера SQL или базы данных SQL на главной ЭВМ.

1.3.1. PostgreSQL

PostgreSQL - (произносится «Пост- Грес -Кью-Эл» или просто «постгрес») — это свободно распространяемая объектно -реляционная система управления базами данных (ORDBMS), наиболее развитая

из открытых СУБД в мире и являющаяся реальной альтернативой коммерческим базам данных (таким как Oracle Database, Microsoft SQL Server, IBM DB2, Informix и СУБД производства Sybase) вместе с другими свободными СУБД (такими как MySQL и Firebird).

PostgreSQL поддерживается на всех современных Unix системах (34 платформы), включая наиболее распространенные, такие как Linux, FreeBSD, NetBSD, OpenBSD, SunOS, Solaris, DUX, а также под Mac OS X. Начиная с версии 8.X PostgreSQL работает в "native" режиме под MS Windows NT, Win2000, WinXP, Win2003. Известно, что есть успешные попытки работать с PostgreSQL под Novell Netware 6 и OS2.

Основные характеристики:

1) Надежность PostgreSQL является проверенным и доказанным фактом и обеспечивается следующими возможностями:

- полное соответствие принципам ACID (Atomicity Consistency Isolation Durability) - атомарность, непротиворечивость, изолированность, сохранность данных.

- многоверсионность (Multiversion Concurrency Control, MVCC) используется для поддержания согласованности данных в конкурентных условиях, в то время как в традиционных базах данных используются блокировки.

- наличие Write Ahead Logging (WAL) - общепринятый механизм протоколирования всех транзакций, что позволяет восстановить систему после возможных сбоев.

- Point in Time Recovery (PITR) - возможность восстановления базы данных (используя WAL) на любой момент в прошлом, что позволяет осуществлять непрерывное резервное копирование кластера PostgreSQL.

- целостность данных является сердцем PostgreSQL. Помимо MVCC, PostgreSQL поддерживает целостность данных на уровне схемы - это внешние ключи (foreign keys), ограничения (constraints).

- Модель развития PostgreSQL, которая абсолютно прозрачна для любого, так как все планы, проблемы и приоритеты открыто обсуждаются.

2) Производительность PostgreSQL основывается на использовании индексов, интеллектуальном планировщике запросов, тонкой системы блокировок, системе управления буферами памяти и кэширования, превосходной масштабируемости при конкурентной работе.

3) Расширяемость PostgreSQL означает, что пользователь может настраивать систему путем определения новых функций, агрегатов, типов, языков, индексов и операторов. Объектно-ориентированность PostgreSQL позволяет перенести логику приложения на уровень базы данных, что сильно упрощает разработку клиентов, так как вся бизнес логика находится в базе данных. Функции в PostgreSQL однозначно определяются названием, количеством и типами аргументов.

4) Поддержка SQL – поддержка основных возможностей, присущих любой SQL базе данных.

5) Богатый набор типов данных PostgreSQL включает:

- численные типы (целые, с фиксированной точкой, с плавающей точкой, денежный тип [отличается специальным форматом вывода, а в остальном аналогичен числам с фиксированной точкой с двумя знаками после запятой]);

- символьные типы произвольной длины;

- двоичные типы (включая BLOB);

- типы «дата/ время» (полностью поддерживающие различные форматы, точность, форматы вывода, включая последние изменения в часовых поясах);

- булев тип
- перечисление
- геометрические примитивы
- сетевые типы (IP и IPv6-адреса, CIDR-формат, MAC-адрес);
- UUID-идентификатор
- XML-данные
- массивы
- OID-типы
- псевдотипы

Более того, пользователь может самостоятельно создавать новые требуемые ему типы и программировать для них механизмы индексирования с помощью GiST.

б) Безопасность данных также является важнейшим аспектом любой СУБД. В PostgreSQL она обеспечивается 4-мя уровнями безопасности:

- PostgreSQL нельзя запустить под привилегированным пользователем - системный контекст;
- SSL,SSH шифрование трафика между клиентом и сервером - сетевой контекст;
- сложная система аутентификации на уровне хоста или IP адреса/подсети. Система аутентификации поддерживает пароли, зашифрованные пароли, Kerberos, IDENT и прочие системы, которые могут подключаться используя механизм подключаемых аутентификационных модулей;
- детализированная система прав доступа ко всем объектам базы данных, которая совместно со схемой, обеспечивающая изоляцию названий объектов для каждого пользователя, PostgreSQL предоставляет богатую и гибкую инфраструктуру.

Сильными сторонами PostgreSQL считаются:

- поддержка БД практически неограниченного размера, см. таблицу 1;

Таблица 1 – Ограничения PostgreSQL

Максимальный размер БД	Нет ограничений
Максимальный размер таблицы	32 Тбайта
Максимальный размер записи	1.6 Т.Байт
Максимальный размер поля	1 Гбайт
Максимум записей в таблице	Нет ограничений
Максимум полей в таблице	250—1600, в зависимости от типов полей
Максимум индексов в таблице	Нет ограничений

- мощные и надёжные механизмы транзакций и репликации;
- наследование - таблицы могут наследовать характеристики и наборы полей от других таблиц (родительских). При этом данные , добавленные в порождённую таблицу, автоматически будут участвовать (если это не указано отдельно) в запросах к родительской таблице.
- легкая расширяемость – на базе PostgreSQL компанией EnterpriseDB созданы более мощные варианты этой СУБД, являющиеся платными для коммерческого использования.

1.4. Обзор языков и средств, использованных работе

1.4.1. C++

C++ - компилируемый, статистически -типизированный язык, предназначенный для программирования общего назначения.

Поддерживает такие парадигмы программирования, как процедурное программирование, объектно-ориентированное программирование, обобщённое программирование. Язык имеет богатую стандартную библиотеку, которая включает в себя распространённые контейнеры и алгоритмы, ввод-вывод, регулярные выражения, поддержку многопоточности и другие возможности. C++ сочетает свойства как высокоуровневых, так и низкоуровневых языков. В сравнении с его предшественником — языком C, — наибольшее внимание уделено поддержке объектно-ориентированного и обобщённого программирования.

C++ широко используется для разработки программного обеспечения, являясь одним из самых популярных языков программирования. Область его применения включает создание операционных систем, разнообразных прикладных программ, драйверов устройств, приложений для встраиваемых систем, высокопроизводительных серверов, а также развлекательных приложений (игр). Существует множество реализаций языка C++, как бесплатных, так и коммерческих и для различных платформ. Например, на платформе x86 это GCC, Visual C++, Intel C++ Compiler, Embarcadero (Borland) C++ Builder и другие. C++ оказал огромное влияние на другие языки программирования, в первую очередь на Java и C#.

Синтаксис C++ унаследован от языка C. Одним из принципов разработки было сохранение совместимости с C. Тем не менее, C++ не является в строгом смысле надмножеством C; множество программ,

которые могут одинаково успешно транслироваться как компиляторами С, так и компиляторами С++, довольно велико, но не включает все возможные программы на С.

1.4.2. PHP

PHP — скриптовый язык общего назначения, интенсивно применяемый для разработки веб -приложений. В настоящее время поддерживается подавляющим большинством хостинг-провайдеров и является одним из лидеров среди языков , применяющихся для создания динамических веб-сайтов<https://ru.wikipedia.org/wiki/PHP> - cite_note-tiobe-8.

Язык и его интерпретатор (Zend Engine) разрабатываются группой энтузиастов в рамках проекта с открытым кодом. Проект распространяется под собственной лицензией, несовместимой с GNU GPL.

В области веб-программирования, в частности серверной части, PHP — один из популярных сценарных языков (наряду с JSP, Perl и языками, используемыми в ASP.NET).

Популярность в области построения веб-сайтов определяется наличием большого набора встроенных средств для разработки веб-приложений [10].

Основные из них:

- автоматическое извлечение POST и GET-параметров, а также переменных окружения веб-сервера в предопределённые массивы;
- взаимодействие с большим количеством различных систем управления базами данных (MySQL, MySQLi, SQLite, PostgreSQL, Oracle (OCI8), Oracle, Microsoft SQL Server, Sybase, ODBC, mSQL, IBM DB2, Cloudscape и Apache, Derby, Informix, Ovrimos, SQL, LotusNotes, DB++, DBM, dBase, DBX, FrontBase, FilePro, Ingres II, SESAM, Firebird / InterBase)
- автоматизированная отправка HTTP-заголовков;
- работа с HTTP-авторизацией;
- работа с cookies и сессиями;

- работа с локальными и удалёнными файлами, сокетами;
- обработка файлов, загружаемых на сервер;
- работа с XForms.

В настоящее время PHP используется сотнями тысяч разработчиков . Согласно рейтингу корпорации TIOBE, базирующемся на данных поисковых систем, в мае 2016 года PHP находился на 6 месте среди языков программирования:. К крупнейшим сайтам , использующим PHP, относятся Facebook, Wikipedia и др.

Входит в LAMP — распростран ённый набор программного обеспечения для создания и хостинга веб- сайтов (Linux, Apache, MySQL, PHP).

1.5. Обзор контроллеров Arduino Uno

В данный момент на рынке существует большое количество контроллеров которые позволяют создать СКУД , большинство из них имеют собственное ПО , однако множество зарубежных представителей не предоставляют удобную среду для их использования.

В данной работе будет использоваться контроллер Arduino Uno с NFC RC 522 A .(Рис 3)

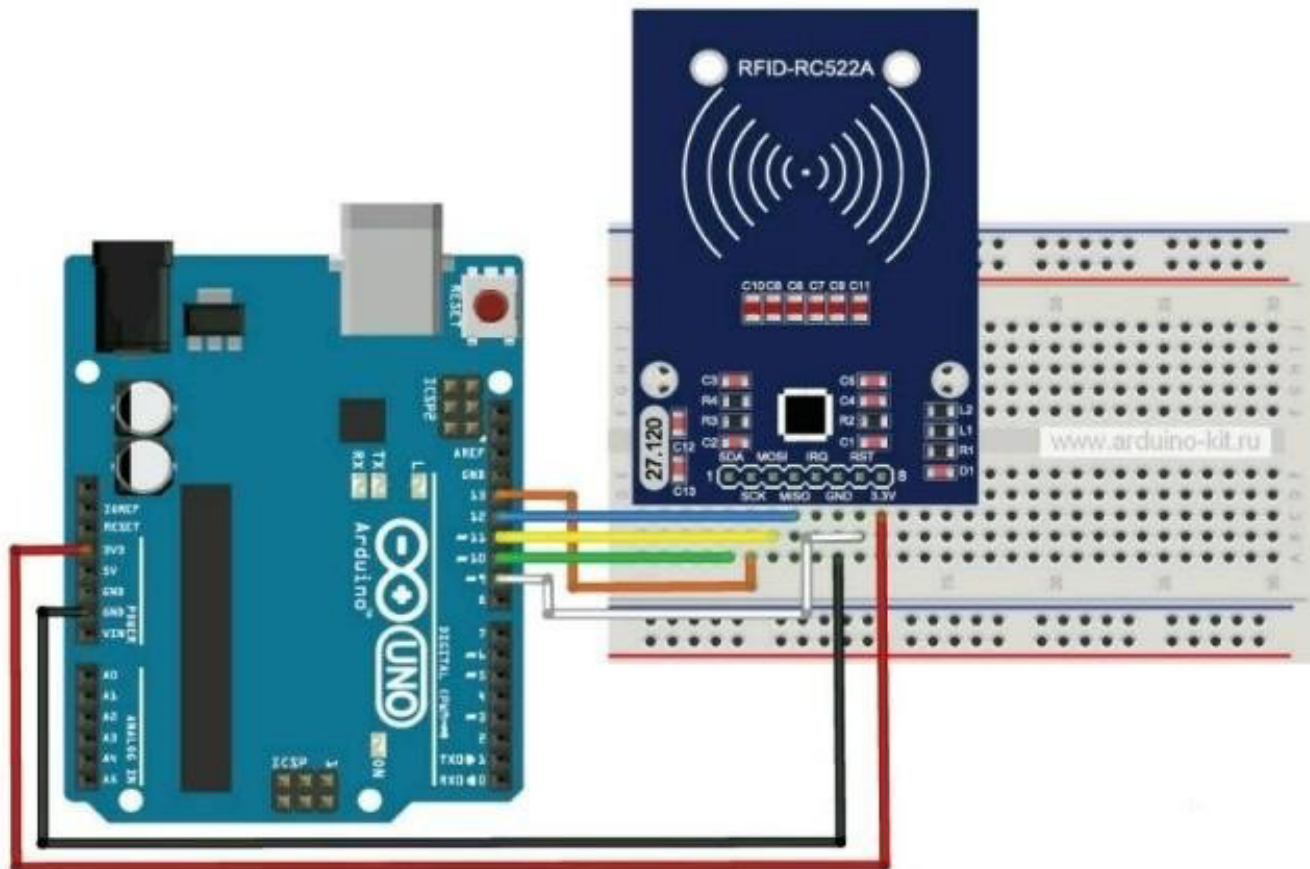


Рис.3. Arduino Uno с NFC RC522A

Изм.	Лист	№ докум.	Подпись	Дата

27.03.04.2018 222.00.00.ПЗ

Приципиальная схема

Arduino™ UNO Reference Design

Reference Design ARE PROVIDED "AS IS" AND "WITH ALL FAULTS". ARDUINO DISCLAIMS ALL OTHER WARRANTIES, EXPRESS OR IMPLIED, REGARDING PRODUCTS, INCLUDING BUT NOT LIMITED TO, ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Arduino may make changes to specifications and product descriptions at any time, without notice. The Customer must not rely on the absence or characteristics of any features or instructions marked "reserved" or "undefined". Arduino reserves the right for future definitions and shall have no responsibility whatsoever for conflicts or incompatibilities arising from future changes to them. The product information on the Web Site or Materials is subject to change without notice. Do not finalize a design with this information.

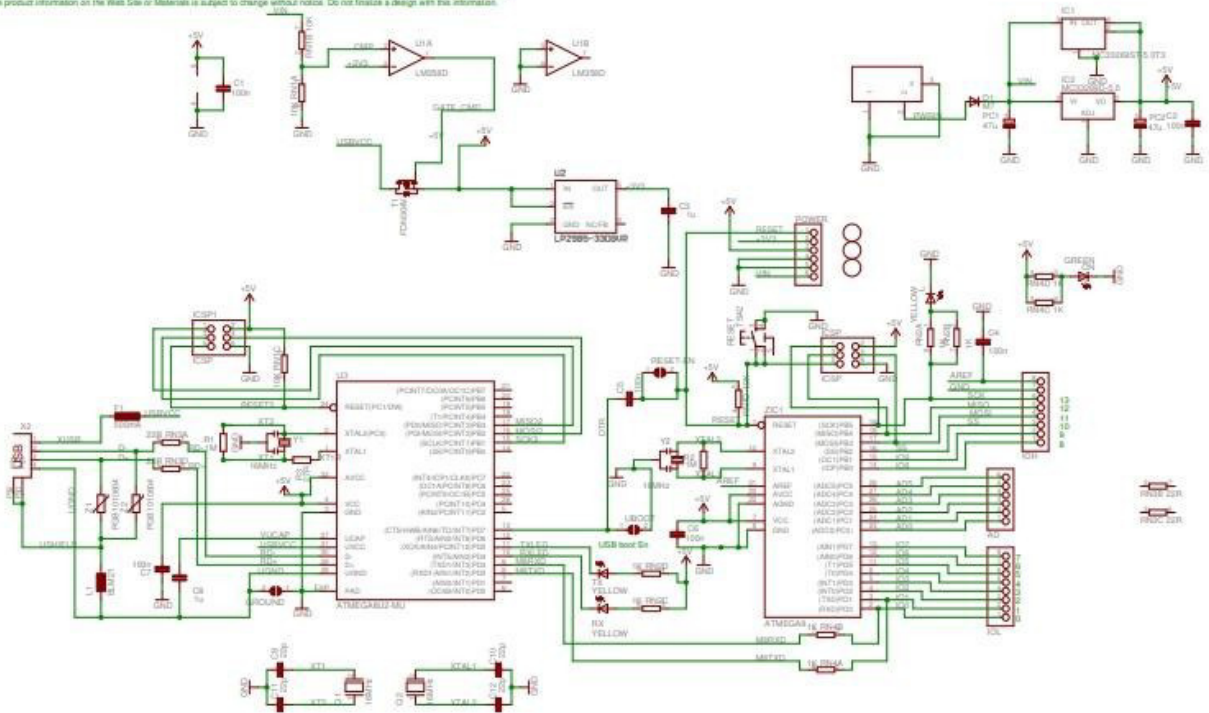


Рис .4. Приципиальная схема Arduino UNO

Характеристики

Микроконтроллер

ATmega328

Рабочее напряжение

5 В

Входное напряжение (рекомендуемое)

7-12 В

Входное напряжение (предельное)

6-20 В

Цифровые Входы/Выходы

14 (6 из которых могут использоваться как выходы ШИМ)

Аналоговые входы

6

Постоянный ток через вход/выход

40 мА

Постоянный ток для вывода 3.3 В

50 мА

Флешпамять

32 Кб (ATmega328) из которых 0.5 Кб используются для загрузчика

ОЗУ

2 Кб (ATmega328)

EEPROM

1 Кб (ATmega328)

Тактовая частота

16 МГц

Изм.	Лист	№ докум.	Подпись	Дата

27.03.04.2018 222.00.00.ПЗ

Лист

41

Общие сведения

Arduino Uno контроллер построен на ATmega328. Платформа имеет 14 цифровых вход / выходов (6 из которых могут использоваться как выходы ШИМ), 6 аналоговых входов, кварцевый генератор 16 МГц, разъем USB, силовой разъем, разъем ICSP и кнопку перезагрузки. Для работы необходимо подключить платформу к компьютеру посредством кабеля USB, либо подать питание при помощи адаптера AC/DC или батареи.

Питание

Arduino Uno может получать питание через подключение USB или от внешнего источника питания. Источник питания выбирается автоматически.

Внешнее питание (не USB) может подаваться через преобразователь напряжения AC/DC (блок питания) или аккумуляторной батареей. Преобразователь напряжения подключается посредством разъема 2.1 мм с центральным положительным полюсом. Провода от батареи подключаются к выводам Gnd и Vin разъема питания.

Платформа может работать при внешнем питании от 6 В до 20 В. При напряжении питания ниже 7 В, вывод 5V может выдавать менее 5 В, при этом платформа может работать нестабильно. При использовании напряжения выше 12 В регулятор напряжения может перегреться и повредить плату. Рекомендуемый диапазон от 7 В до 12 В.

Выводы питания:

- VIN. Вход используется для подачи питания от внешнего источника (в отсутствие 5 В от разъема USB или другого регулируемого источника питания). Подача напряжения питания происходит через данный вывод.
- 5V. Регулируемый источник напряжения, используемый для питания микроконтроллера и компонентов на плате. Питание может подаваться от вывода VIN через регулятор напряжения, или от разъема USB, или другого регулируемого источника напряжения 5 В.
- 3V3. Напряжение на выводе 3.3 В генерируемое встроенным регулятором на плате. Максимальное потребление тока 50 мА.
- GND. Выводы заземления.

					27.03.04.2018 222.00.00.ПЗ	Лист
Изм.	Лист	№ докум.	Подпись	Дата		42

Связь

На платформе Arduino Uno установлено несколько устройств для осуществления связи с компьютером, другими устройствами Arduino или микроконтроллерами. ATmega328 поддерживают последовательный интерфейс UART TTL (5 В), осуществляемый выводами 0 (RX) и 1 (TX). Установленная на плате микросхема ATmega8U2 направляет данный интерфейс через USB, программы на стороне компьютера "общаются" с платой через виртуальный COM порт. Прошивка ATmega8U2 использует стандартные драйвера USB COM, никаких сторонних драйверов не требуется, но на Windows для подключения потребуется файл ArduinoUNO.inf. Мониторинг последовательной шины (Serial Monitor) программы Arduino позволяет посылать и получать текстовые данные при подключении к платформе. Светодиоды RX и TX на платформе будут мигать при передаче данных через микросхему FTDI или USB подключение (но не при использовании последовательной передачи через выводы 0 и 1).

Библиотекой SoftwareSerial возможно создать последовательную передачу данных через любой из цифровых выводов Uno.

ATmega328 поддерживает интерфейсы I2C (TWI) и SPI. В Arduino включена библиотека Wire для удобства использования шины I2C.

Программирование

Платформа программируется посредством ПО Arduino. Из меню Tools > Board выбирается «Arduino Uno» (согласно установленному микроконтроллеру). Подробная информация находится в справочнике и инструкциях.

Микроконтроллер ATmega328 поставляется с записанным загрузчиком, облегчающим запись новых программ без использования внешних программаторов. Связь осуществляется оригинальным протоколом STK500.

Токовая защита разъема USB

В Arduino Uno встроен самовостанавливающийся предохранитель (автомат), защищающий порт USB компьютера от токов короткого замыкания и сверхтоков. Хотя практически все компьютеры имеют подобную защиту, тем не менее, данный предохранитель обеспечивает дополнительный

					27.03.04.2018 222.00.00.ПЗ	Лист
Изм.	Лист	№ докум.	Подпись	Дата		43

барьер. Предохранитель срабатывает при прохождении тока более 500 мА через USB порт и размыкает цепь до тех пор пока нормальные значения токов не будут восстановлены.

РАЗРАБОТКА СИСТЕМЫ АДМИНИСТРИРОВАНИЯ СКУД

2.1. Составление блок-схемы работы системы

Для того чтобы понять взаимодействие всех узлов системы СКУД понадобится блок-схема взаимодействия элементов СКУД в которую будут входить:

- датчик входа;
- датчик выхода;
- Контроллер Arduino Uno;
- Сервер.

2.2. Програмное решение чтения меток.

Так как наша метка имеет уникальный идентификатор, нам нужно извлечь его из неё для этого применяем:

```
void exercise(char32_t type, NfcTag tag){  
    Serial.println(tag.getTagType());  
    Serial.print("UID: ");  
    Serial.println(tag.getUidString());    // Извлекает уникальный  
идентификатор из вашей метки
```

```

        if(tag.getUidString() !=
        Serial.println("connecting...");
    if (client.connect(server, 80)) {
        Serial.println("connected");
        client.println("GET/nfc/?q=arduino&type="+type+"&nfc="+tag.getUidString()+
"&pas=secret HTTP/1.0");
        client.println();
    } else
    {
        Serial.println("connection failed");
    }

```

В результате мы получаем общения контроллера с сервером и проверкой уникального идентификатора (кода) вещественного носителя.

В наше программе метки имеют разделение так, одна метка осуществляет проверку и запись данных о входе, а вторая данных о выходе сотрудника.

```

void loop()
{
    if (nfc_out.tagPresent()){
        exercise("out", nfc_out.read());
    }
    if (nfc_in.tagPresent()){
        exercise("in", nfc_in.read());
    }
}

```

Данный метод позволяет без труда определить в какую именную часть БД нужно внести данные т.е. приход или уход работника.

Так же программа подаёт сигнал для открытия двери (или турникета).

```

if (client.available()) {

```

```

        char c = client.read();
        if(c == "true") {
    openDoor();
        }
    }
    client.stop();
    client.flush();
}
    delay(10000);
}
void openDoor(){
    //открываем дверь
}

```

Проводим инициализацию всех контроллеров , устанавливаем задержки и инициализируем подключение к сети.

```

byte mac[] = { 0xDE, 0xAD, 0xBE, 0xEF, 0xFE, 0xED }; // Мак адрес
PN532_I2C pn532_i2c(Wire);
NfcAdapter nfc_in = NfcAdapter(pn532_i2c); // Указываем , что
используем плату расширения
PN536_I2C pn536_i2c(Wire);
NfcAdapter nfc_out = NfcAdapter(pn536_i2c); // Указываем, что
используем плату расширения
    EthernetClient client;
void setup()
{
    //Старт
    Serial.begin(9600);
    Ethernet.begin(mac);
    nfc_out.begin();

```

```
nfc_in.begin();
  delay(1000);
}
```

Контроллер постоянно находится в режиме считывания .

```
void loop()
{
  if (nfc_out.tagPresent()){
    exercise("out", nfc_out.read());
  }
  if (nfc_in.tagPresent()){
    exercise("in", nfc_in.read());
  }
}
```

После того как на метку пришёл сигнал идёт запрос к серверу на проверку данных. В случае ответа «true» вызывается метод открытия дери.

```
void exercise(char32_t type, NfcTag tag){
  Serial.println(tag.getTagType());
  Serial.print("UID: ");
  Serial.println(tag.getUidString()); // Извлекает уникальный
идентификатор из вашей метки
  if(tag.getUidString() != ""){
    Serial.println("connecting...");
    if (client.connect(server, 80)) {
      Serial.println("connected");
      client.println("GET
/nfc/?q=arduino&type="+type+"&nfc="+tag.getUidString()+"&pas=secret
HTTP/1.0");
```

```

        client.println();
    } else {
        Serial.println("connection failed");
    }

    if (client.available()) {
        char c = client.read();
        if(c == "true") {
            openDoor();
        }
    }
    client.stop();
    client.flush();
}
delay(10000);
}

```

2.5. Требования к информационной безопасности программной системы

Развитие средств, методов и форм автоматизации процессов хранения и обработки информации и массовое применение персональных компьютеров делают информацию гораздо более уязвимой. Информация, циркулирующая в них, может быть незаконно изменена, похищена или уничтожена. Поэтому на сегодняшний день системы, содержащие конфиденциальные данные, нуждаются в комплексном обеспечении информационной безопасности.

Первым самым важным требованием к информационной безопасности программной системы является устранение возможности подбора паролей, что позволит устранить несанкционированный доступ к данным, их хищение и изменение [22]. Разработанная система должна отслеживать

каждую попытку авторизации пользователя и , в случае чрезмерной активности, блокирует пользователя на определенное время.

Вторым важным требованием безопасности является защита от внедрения SQL-кода, т.к. это один из распространенных способов взлома сайтов и программ, работающих с базами данных, который основывается на внедрении в запрос произвольного SQL-кода [23]. Внедрение SQL может дать возможность атакующему выполнить произвольный запрос к базе данных (например , прочитать содержимое любых таблиц, удалить, изменить или добавить данные), получить возможность чтения и/или записи локальных файлов и выполнения произвольных команд на атакуемом сервере. В разработанной программной системе для предотвращения подобного вида атак должна быть реализована фильтрация входные параметров , значения которых будут использованы для построения SQL-запроса, а именно: фильтрация строковых параметров, целочисленных параметров, усечение входных параметров.

Третьим важным требованием является разграничение пользователей на группы, как на уровне базы данных, так и на уровне системы [23]. На уровне базы данных должны создаваться пользователи с определенными привилегиями, которые позволяют соединиться с базой данных (создавать сессию), выбирать строки таблиц , вставлять данные в таблицу. На уровне системы каждому типу пользователей должна предоставляться собственная защищенная область, которая позволяет не выходить за рамки определенных возможностей, для начала примем две вариации администраторов (1) и пользователей (0).

2.3.Серверная часть

Так же нельзя забывать о вариантах развития, система является масштабируемой, поэтому в дальнейшем можно ограничить уровень допуска к различным помещениям.

Плата Arduino посылает нам два вида данных, с датчика входа и с датчика выхода выборку осуществляет функция:

```
 ".$_REQUEST["type"]."
```

Позволяет системе определить с какого из двух датчиков был получен запрос.

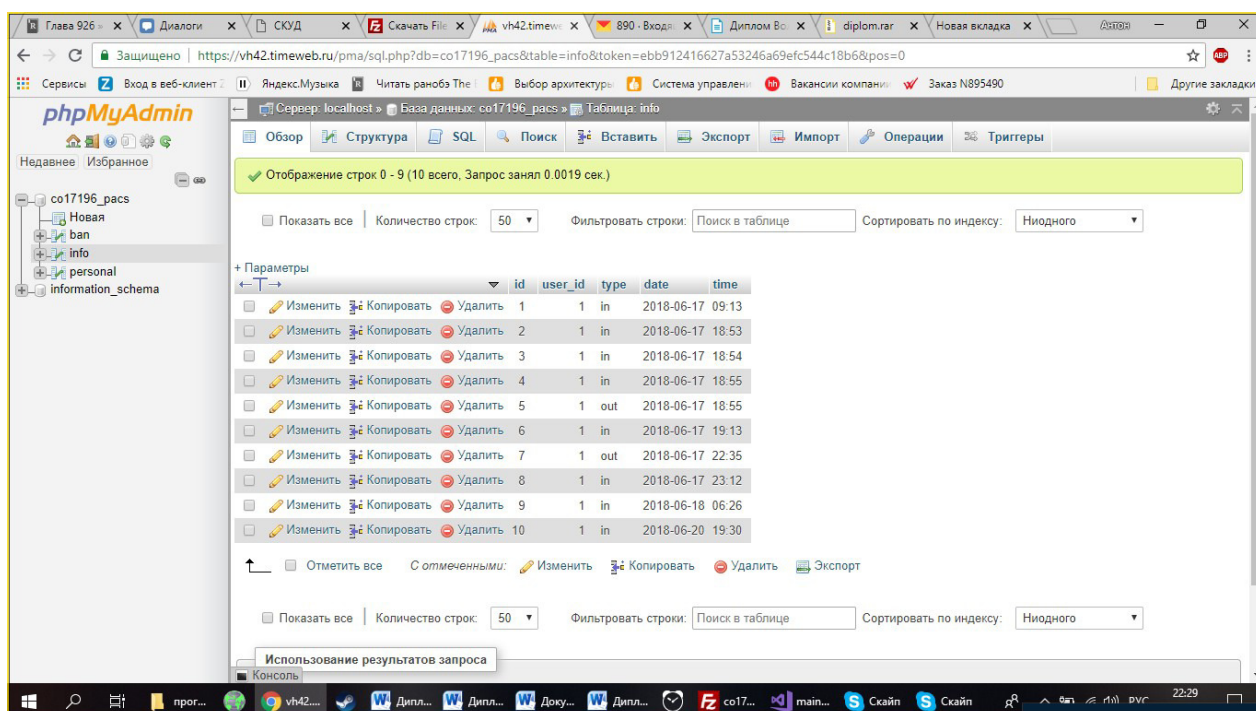


Рис 6.БД входы, выходы.

В нашей системе будет учитываться только данные входа или выхода, в дальнейшем развитие так же можно будет учитывать данные с какой именно точки на предприятие пришли данные.

2.4.Интерфейс

При проектировании интерфейса веб-сайта было решено следовать минимализму и использовать только те элементы, которых будет

достаточно для реализации поставленной задачи. Также внимание было уделено тому, чтобы разрабатываемый интерфейс был интуитивно понятен пользователям и специального обучения по работе с программной системой.

При открытии мы увидим стандартное окно с требованием ввести индивидуальный логин и пароль.(рис.7) Код это странице представлен в приложении 3 .

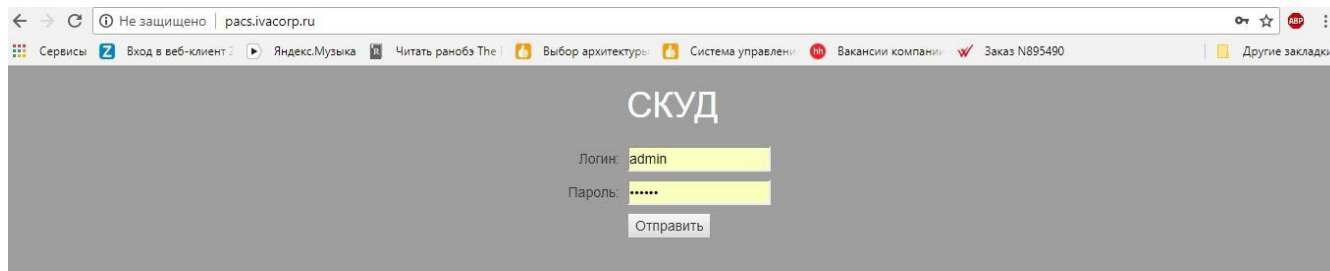


Рис.7. Начальный экран.

При правильной паре логин пароль переходим на следующую страницу на которой нам предложен ряд функций(рис.8).

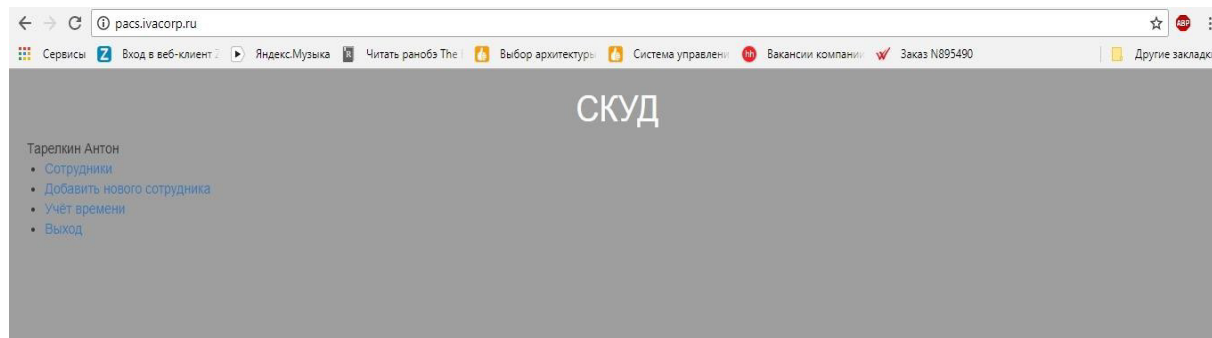


Рис. 8 Функции

В зависимости от различных прав доступа нам будет доступно различное количество функционала. Сотрудники не имеющие прав администратора, могут просматривать только своё время, и список работников, но не могут добавлять новых сотрудников. Код содержится в приложении 4.

2.4.1. Описание основных алгоритмов

Алгоритм: «блокировка пользователя по ip» (рис. 9).

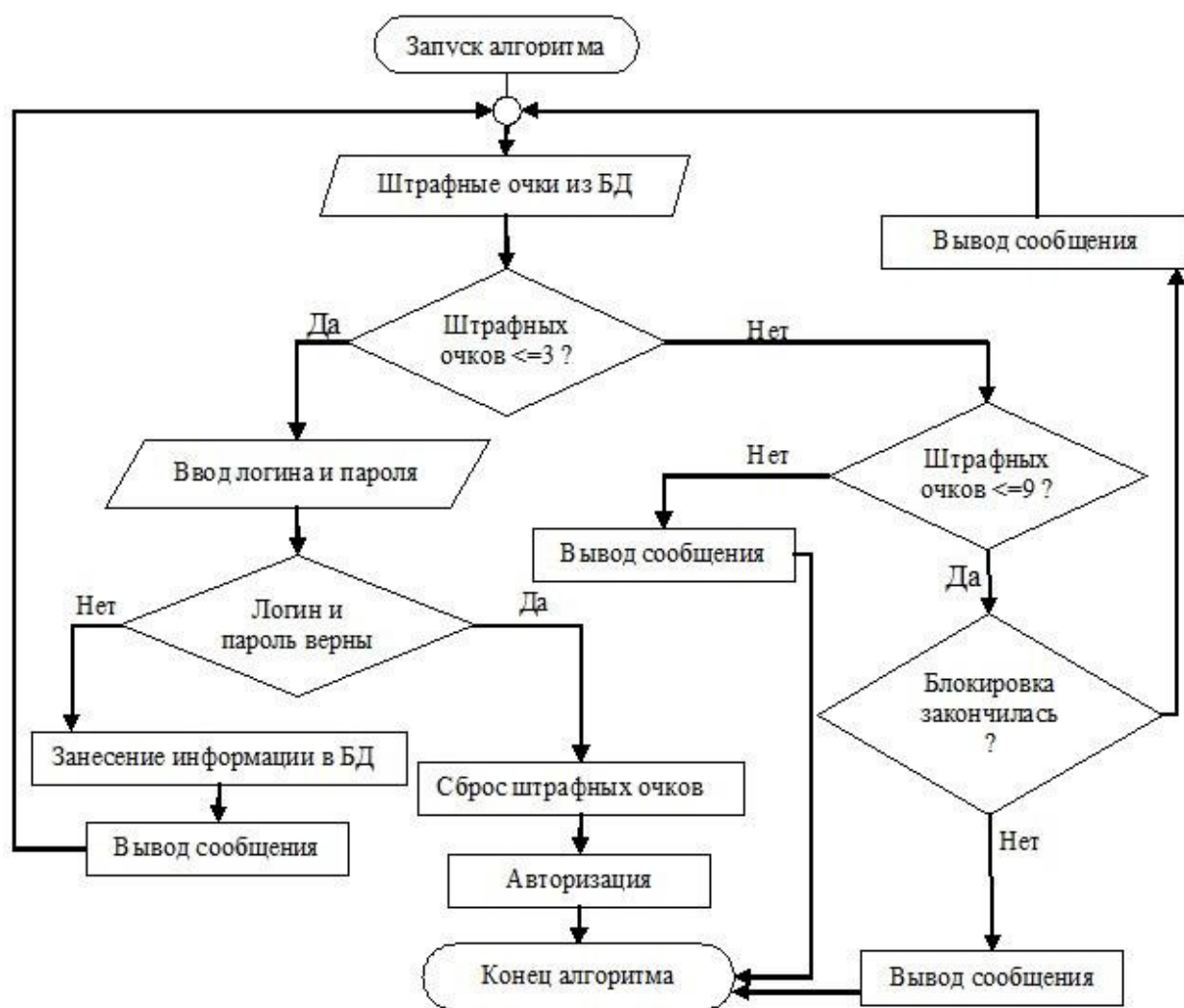


Рис.9 Алгоритм

Если пользователь несколько раз совершает неудачную авторизацию на сайте, то система за такие попытки начисляет пользователю штрафные очки. Если пользователь набрал от 3 до 10 штрафных очков, то система блокирует пользователя на некоторое время, которое зависит от уже набранных штрафных очков (по 20 минут за каждую попытку неправильного ввода) и предлагает подождать и вспомнить пароль, по истечении блокировки пользователь вновь сможет повторить попытку авторизации. При удачной попытке авторизации, если пользователь не набрал еще больше 10 штрафных очков, они с него списываются. Если пользователь набирает больше 10 штрафных очков, ему приходит сообщение «для разблокировки обратитесь к администратору системы».

(рис. 10)

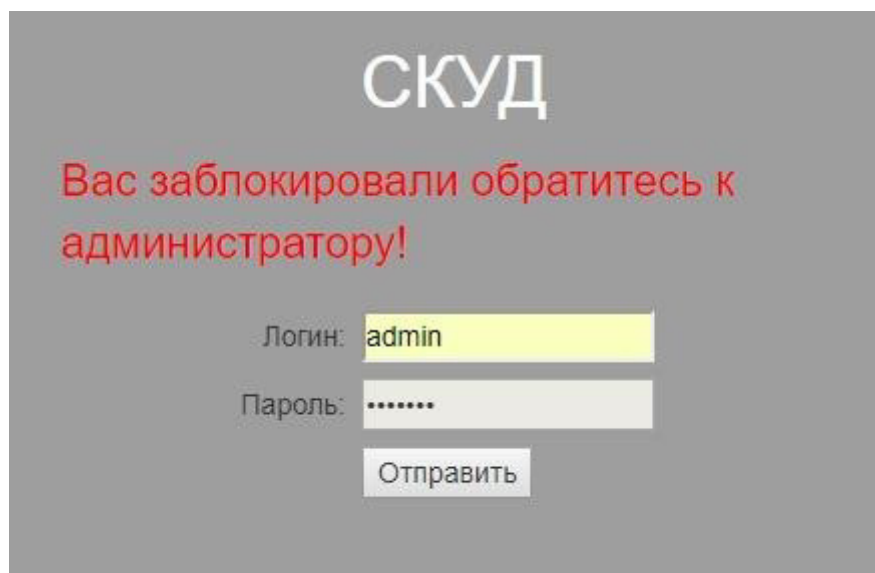


Рис.10.Сообщение об ошибке

Алгоритм учёта времени

Данный алгоритм делает выборку всех данных по выбранному дню конкретного сотрудника, и производит подсчёт времени по нахождению сотрудника на рабочем месте.

Данные по входу и выходу за день суммируется, в итоге мы получаем полное время с учётом всех входов и выходов сотрудника за день.

```
if($_SESSION['G_class_ses']==1)
    $query = "select `type`, `time` from info where
`date`='".$_$_REQUEST["Y"]."-".$_$_REQUEST["m"]."-".$_$_REQUEST["d"]."' and
`user_id`='".$_$_REQUEST["id"]."';";
    else {
        $query = "select `type`, `time` from info where
`date`='".$_$_REQUEST["Y"]."-".$_$_REQUEST["m"]."-".$_$_REQUEST["d"]."' and
`user_id`='".$_$_SESSION['myID']."';";
    }
$result = mysql_query($query);
$type = "in";
$count_time = array(0,0);
$temp_time = "";
while($row = mysql_fetch_array($result)):
    if($row['type'] == "in" && $type == "in") {
        $temp_time = $row["time"];
        $type = "out";
    }
    if($row['type'] == "out" && $type == "out") {
        $type = "in";
        $date1 = new DateTime($temp_time);
        $date2 = new DateTime($row["time"]);
        $diff = $date2->diff($date1);
        $count_time[0] += $diff->format('%h');
        if($count_time[1]+$diff->format('%i')>59) {
            $count_time[1] += $diff->format('%i');
            $count_time[1] -= 60;
        }
    }
}
```


комбинации данных пользовательского уровня, непредусмотренные сценарии использования, отсутствующая или неверная функциональность, неудобство использования.

Функциональное тестирование – это тестирование программного обеспечения в целях проверки реализуемости функциональных требований, то есть способности программного обеспечения в определенных условиях решать задачи, нужные пользователям. Функциональные требования определяют, что именно делает программное обеспечение, какие задачи оно решает. Все тесты на функциональное тестирование прошли успешно.

Тестирование безопасности системы. Для тестирования безопасности системы был произведен тест модуля «Авторизация» и прямой доступ к защищенным областям сайта. Тест модуля «Авторизация» показал, что происходит полная и частичная блокировка пользователя при многократном неправильном вводе логина и пароля. Сообщения, выводимые пользователю, соответствовали ожидаемым.

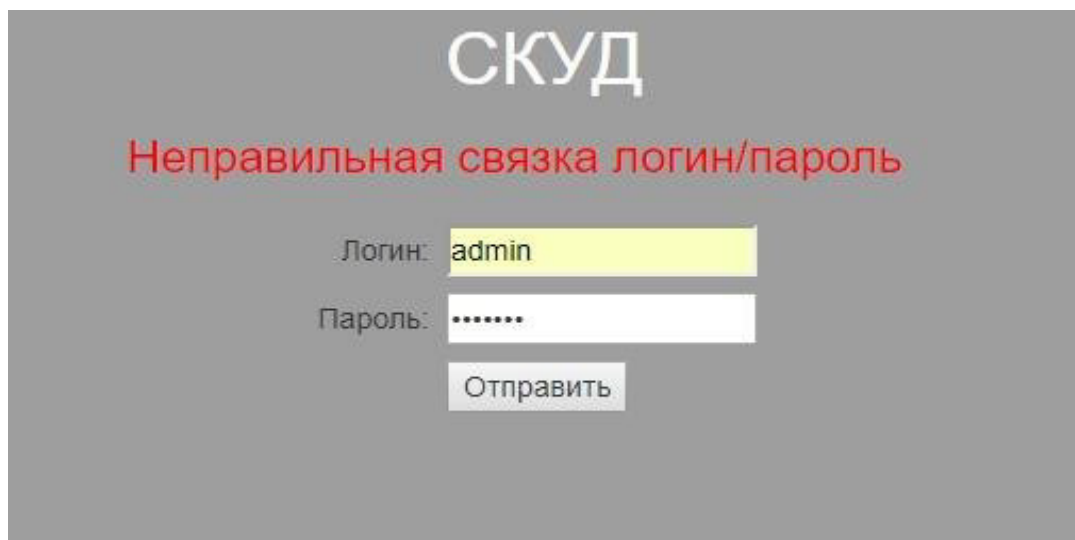


Рис. 13 Результат теста.

2.6. Недостатки системы и возможные варианты улучшений

В ходе тестирования были выделены незначительные недостатки системы, которые позволили определить поле для дальнейшей работы и улучшения.



+ Параметры			id	user_id	type	date	time
<input type="checkbox"/>		Изменить	1	1	in	2018-06-17	09:13
<input type="checkbox"/>		Изменить	2	1	in	2018-06-17	18:53
<input type="checkbox"/>		Изменить	3	1	in	2018-06-17	18:54
<input type="checkbox"/>		Изменить	4	1	in	2018-06-17	18:55
<input type="checkbox"/>		Изменить	5	1	out	2018-06-17	18:55
<input type="checkbox"/>		Изменить	6	1	in	2018-06-17	19:13
<input type="checkbox"/>		Изменить	7	1	out	2018-06-17	22:35
<input type="checkbox"/>		Изменить	8	1	in	2018-06-17	23:12
<input type="checkbox"/>		Изменить	9	1	in	2018-06-18	06:26
<input type="checkbox"/>		Изменить	10	1	in	2018-06-20	19:30

Один из них – отсутствие реакции системы на то, что пользователь может выполнить вход несколько раз в течение дня, при этом, не отметившись на выходе ни разу. Требуется доработка для передачи данных администратору (назначенному лицу) о нарушении, таким образом, администратором должна производиться корректировка данны.

Итоги работы

В результате выполнения работы получили рабочее приложение для администрирование системы СКУД , был реализован алгоритм защиты от несанкционированного доступа, был разработан метод подключение контроллера к нашей системе, присоединена база данных, а так же определены слабые места системы, что оставляет нам поле для дальнейшей работы . Были добавлены не только средства слежения за временем работы сотрудников, а так же средства для дальнейшего добавления новых сотрудников в систему.

Таким образом, разработка является полностью рабочим прототипом, который может самостоятельно развиваться, не требуют больших затрат.

Список использованных источников

Законы и иные нормативные акты

1. ГОСТ Р 54831-2011 « Системы контроля и управления доступом. Устройства преграждающие управляемые. Общие технические требования. Методы испытаний» (утв. и введен в действие Приказом Росстандарта от 13.12.2011 № 1223-ст).

2. Волковицкий, В. Д ., Волхонский, В. В . Системы контроля и управления доступом /В.Д. Волковицкий, В.В. Волхонский. – СПб.: Университет ИТМО, 2015. –53 с.

3. Ворона, В.А., Тихонов, В.А. Системы контроля и управления доступом / В.А. Ворона, В.А. Тихонов. –М.: Горячая линия Телеком, 2015. – 272 с.

					27.03.04.2018 222.00.00.ПЗ	Лист
Изм.	Лист	№ докум.	Подпись	Дата		59

4. Джхунян, В.Л., Шаньгин, В.Ф. Электронная идентификация / В.Л. Джхунян, В.Ф. Шаньгин. –М.: АСТ, НТ Пресс 2014. – 696 с.
5. Кашкаров, А.П. Системы безопасности и устройства кодового доступа. Просто о сложном / А.П. Кашкаров. –М.: ДМК Пресс, 2014. 109 с.
6. Мельников, В., Клейменов, С., Петраков, А. Информационная безопасность и защита информации, 2-ое издание / В. Мельников, С. Клейменов, А. Петраков. –М.: Академия, 2008. –336 с.
7. Низатмудинов, М. Тактика защиты и нападения на WEB-приложения / М. Низатмудинов. – СПб.: БХВ-Петербург, 2005. – 432 с.
8. Программно-аппаратный комплекс «Менуэт 2000». Руководство по эксплуатации. –М.: Москва, 1999. –118 с.
9. Рыжова, В.А. Проектирование и исследование комплексных систем безопасности / В.А. Рыжова. –СПб.: НИУИТМО, 2012. –157с.

Статьи и научные публикации

10. Сорокин, К. Применение биометрических технологий в обеспечении информационной безопасности бизнеса. / К. Сорокин // СКУД. Антитерроризм-2013. –2013. –С. 46-47.
11. Функции универсальных СКУД: что нужно потребителю / О.О. Тихонов, А.С. Малышева, А.В. Шаповалов, А.Е. Гамбург, Л.А. Стасенко, А.С. Курилин //Системы безопасности. –2011. –№ 4. – С . 108-119.

Электронные ресурсы

					27.03.04.2018 222.00.00.ПЗ	Лист
Изм.	Лист	№ докум.	Подпись	Дата		60

12. Веб-системы контроля доступа. – URL: <http://www.security.honeywell.com/ru/products/access/so/779994.html/> (дата обращения: 16.05.2018).

13. Интегрированная система охраны «Орион». – URL: <https://bolid.ru/production/orion/> (дата обращения: 26.05.2018)

14. Масштабируемые системы контроля доступа и интегрированные системы безопасности Lenel OnGuard. – URL: <http://www.lenel.ru/news/access-control-system.ahtm/> (дата обращения: 16.05.2016).

15. Министерство по управлению государственным имуществом Свердловской области. – URL: http://mugiso.midural.ru/region/okn/estateekt.php?ELEMENT_ID=949/ (дата обращения: 19.05.2018).

16. Многофункциональный центр предоставления государственных и муниципальных услуг. – URL: http://www.mfc66.ru/news/8228/?sphrase_id=57566/ (дата обращения: 19.05.2018).

17. Не типичные функции СКУД. – URL: http://www.secuteck.ru/articles2/sys_ogr_dost/netipichnye-funktsii-skud/ (дата обращения: 20.05.2018).

18. Обзор возможностей СКУД . – URL: <http://www.sistema-dostupa.ru/i03.htm/> (дата обращения: 16.05.2018).

19. Система контроля доступа на предприятии. Особенности внедрения. – URL: <http://www.cleper.ru/articles/description.php?n=441> (дата обращения: 20.05.2018).

20. Система контроля и управления доступом. Принцип действия. – URL: <http://www.intersyst.ru/solutions/165/460/> (дата обращения: 14.05.2018).

Приложение 1. Shield

<?

```
    //$its_access[0] = 0;
    if (!$G_isset_session){ header("Location: /"); exit; }
    else
    {
        $query = "select `password`, `stut`, `class`, `uid` from personal where id =
        '$_SESSION['myID'].''";
        $result = mysql_query($query);
        $row = mysql_fetch_array($result);

        $type = $row[2];
        $_SESSION['G_class_ses'] = $type;

        if($its_access[$type] == 1 && $row[0] == $_SESSION['Pas'] && $row[1] == 1 &&
        $row[3] == $_SESSION['UID']){}
        else
        {
```

| | | | | | | |
|------|------|----------|---------|------|----------------------------|------|
| | | | | | 27.03.04.2018 222.00.00.ПЗ | Лист |
| Изм. | Лист | № докум. | Подпись | Дата | | 62 |

```

        if($row[3] != $_SESSION['UID']) { // под вашим логином кто-то зашёл,
нарушены параметры безопасности!!!
            header("Location: /logout.php?uin=error");
            exit;
        }

        if($row[1] != 1 ) { // учётка заблокирована
            header("Location: /logout.php?stut=error");
            exit;
        }

        if($row[0] != $_SESSION['Pas']) { // сменили пароль
            header("Location: /logout.php");
            exit;
        }

        if($its_access[$type] != 1 ) { // тебе не доступна эта страница
            header("Location: /");
            exit;
        }
    }
}

?>

```

Приложение 2 . MiniShield

```

<?php

    $query = "select `password`,`stut`,`class`,`uid` from personal where stut = '1' and id =
    '".$_SESSION['myID']."'";
    $result = mysql_query($query);
    $row = mysql_fetch_array($result);

    $type = $row[2];
    $_SESSION['G_class_ses'] = $type;

?>

```

Приложение 3.

```

<?php

    $query = "select `password`,`stut`,`class`,`uid` from personal where stut = '1' and id =
    '".$_SESSION['myID']."'";
    $result = mysql_query($query);
    $row = mysql_fetch_array($result);

    $type = $row[2];
    $_SESSION['G_class_ses'] = $type;

?>

```

Приложение 4.

```

<?

```

| | | | | | | | |
|------|------|----------|---------|------|--|----------------------------|------|
| | | | | | | 27.03.04.2018 222.00.00.ПЗ | Лист |
| Изм. | Лист | № докум. | Подпись | Дата | | | 63 |

```
if(empty($_REQUEST["q"]) || empty($_REQUEST["type"]) || empty($_REQUEST["nfc"]) ||
empty($_REQUEST["pas"]) || $_REQUEST["q"] != "arduino" || $_REQUEST["pas"] != "secret" )
    return false;

if($_REQUEST["type"] != "in" && $_REQUEST["type"] != "out")
    return false;

/*-----
Можно добавить проверку на время работы, и не открывать дверь
-----*/

require $_SERVER["DOCUMENT_ROOT"].'/service/start.php';

$query = "select `id` from personal where stut = '1' and nfc = '".$_REQUEST["nfc"]."'";
$result = mysql_query($query);
$row = mysql_fetch_array($result);

if(!empty($row)) {
    echo "true";

    $query = "insert into info (`user_id`,`type`,`date`,`time`) values
('".$_row["id"]."', '".$_REQUEST["type"]."', '".date("Y-m-d")."', '".date("H:i")."'");
    $result = mysql_query($query);
}

?>
```