

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
федеральное государственное бюджетное образовательное учреждение высшего образования
«Тольяттинский государственный университет»

Институт математики, физики и информационных технологий

(наименование института полностью)

Кафедра «Прикладная математика и информатика»

(наименование)

02.03.03 Математическое обеспечение и администрирование информационных систем

(код и наименование направления подготовки, специальности)

Мобильные и сетевые технологии

(направленность (профиль)/специализация)

ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА (БАКАЛАВРСКАЯ РАБОТА)

на тему Разработка системы учета рабочего времени сотрудника АО
«ВАЗСИСТЕМ» на основе событий системы контроля и управления
доступом

Обучающийся А.А. Явленин

(Инициалы Фамилия)

(личная подпись)

Руководитель к.т.н., В.С. Климов

(ученая степень (при наличии), ученое звание (при наличии), Инициалы Фамилия)

Консультант А.В. Москалюк

(ученая степень (при наличии), ученое звание (при наличии), Инициалы Фамилия)

Тольятти 2023

Аннотация

Тема выпускной квалификационной работы - «Разработка системы учета рабочего времени сотрудника АО «АВТОВАЗ» на основе событий системы контроля и управления доступом».

Актуальность данной ВКР обусловлена необходимостью улучшения СКУД АО «АВТОВАЗ», что позволит увеличить эффективность работы предприятия и сократить затраты на оплату труда, так как удастся более точно производить учёт рабочего времени.

Объектом исследования является система контроля и управления доступом в АО «АВТОВАЗ».

Предметом исследования является СКУД, позволяющая осуществлять контроль над предоставлением доступа на вход и на выход в АО «АВТОВАЗ».

Цель выпускной квалификационной работы – разработка программного обеспечения, улучшающего работу СКУД на АО «АВТОВАЗ».

Методы исследования – микросервисные архитектуры.

Разработан модуль информационной системы. Его архитектура основана на принципах микросервисной архитектуры.

Результаты бакалаврской работы имеют практический интерес, могут быть рекомендованы разработчикам высоконагруженных распределённых информационных систем.

Выпускная квалификационная работа состоит из 47 страниц текста, 25 рисунков, 3 таблиц и 21 источников.

Abstract

The topic of the graduate qualification work is "Development of the system to record the working time of an employee of JSC "AVTOVAZ" on the basis of the events of the access control and management system".

Relevance of this thesis is due to the need to improve the ACS of JSC "AVTOVAZ", which will increase the efficiency of the enterprise and reduce labor costs, as it will be possible to more accurately record the working time.

The object of the study is the access control and management system in JSC "AvtoVAZ".

The subject of the study is the ACS, which allows to control the provision of access to the entrance and exit in JSC "AVTOVAZ".

The purpose of the graduate qualification work is to develop software that improves the operation of the ACS at JSC "AvtoVAZ".

Research methods - microservice architectures.

An information system module was developed. Its architecture is based on the principles of microservice architecture.

The results of the bachelor's work are of practical interest, can be recommended to developers of high-loaded distributed information systems.

Graduation qualification work consists of 47 pages of text, 25 figures, 3 tables and 21 sources.

Содержание

Введение.....	4
1 Характеристика организации и экосистемы сервисов	8
1.1 Характеристика организации.....	8
1.2 Обзор существующих систем контроля и управления доступом	9
1.3 Анализ системы контроля и управления доступом на АО «ВАЗСИСТЕМ»	13
2 Проектирование системы контроля и управления доступом	19
2.1 Анализ программной реализации СКУД на предприятии.....	19
2.2 Проектирование решения проблемы с точностью и достоверностью данных	25
2.3 Проектирование решения проблемы учёта времени на перерывы при расчёте рабочего времени сотрудника в СКУД	28
3 Реализация системы контроля и управления доступом.....	31
3.1 Выбор средств реализации.....	31
3.2 Реализация системы контроля качества данных	34
3.3 Реализация алгоритма решения проблемы учёта времени на перерывы.....	37
3.4 Тестирование системы.....	39
Заключение	42
Список используемых источников.....	44

Введение

За счёт широкого распространения информационных технологий стало возможно создание систем контроля и управления доступом, которые представляют собой совокупность программно-аппаратных технических средств контроля и средств управления, имеющих целью ограничение и регистрацию входа-выхода объектов (людей, транспорта) на заданной территории через «точки прохода»: двери, ворота, КПП.

Сложно себе представить крупную компанию или предприятие государственного значения, где доступ на территорию будет общедоступным. Такое невозможно, так как в противном случае очевидна серьёзная угроза безопасности данных и имущества. К тому же, без должного контроля входящих и выходящих лиц не будет наглядна динамика того, сколько сотрудников находится на рабочем месте. Чтобы решить данные проблемы, существует СКУД.

Система контроля и управления доступом (СКУД) – необходимая информационная система для любой компании, в которой требуется вести отчётность входа-выхода объектов на свою территорию [1]. Основными задачами таких систем является: ограничение доступа на заданную территорию, идентификацию лица, имеющего доступ на заданную территорию. Кроме того, такие системы позволяют: вести учёт рабочего времени сотрудника, вести расчёт заработной платы сотрудника (при интеграции с системами бухгалтерского учёта), ведение базы персонала и посетителей, интегрироваться с системой безопасности (с системой видеонаблюдения, для совмещения архивов событий систем, передачи системе видеонаблюдения извещений о необходимости стартовать запись, повернуть камеру для записи последствий зафиксированного подозрительного события).

Системы контроля и управления доступом (СКУД) - это программное и аппаратное обеспечение, которое предназначено для

контроля и ограничения доступа к определенным зонам, помещениям, объектам или информации. Они могут использоваться для обеспечения безопасности в офисах, банках, государственных учреждениях, аэропортах, медицинских учреждениях и других местах.

СКУД обычно состоит из следующих компонентов:

1. Контроллер доступа: это устройство, которое устанавливается на входе в зону, которую необходимо ограничить. Оно определяет, имеет ли пользователь доступ к зоне, и позволяет или запрещает проход.
2. Считыватель: это устройство, которое используется для считывания информации с идентификационной карты или бейджа пользователя. Считыватель может быть установлен на входе в зону или внутри зоны для контроля доступа к конкретным объектам.
3. Идентификационные карты или бейджи: это устройства, которые используются для идентификации пользователя. Они могут быть магнитными, с чипом или бесконтактными, и могут содержать информацию о пользователе, такую как его имя, фотография, должность и т.д.
4. Программное обеспечение: это компонент, который управляет работой системы контроля и управления доступом. Он обычно позволяет администраторам системы управлять доступом пользователей, создавать отчеты и мониторить систему на наличие проблем.
5. Камеры наблюдения: это устройства, которые используются для видеонаблюдения за объектом и могут быть интегрированы в систему контроля доступа.

СКУД могут быть реализованы в различных формах, включая стандартные ключи и замки, электронные ключи, биометрические системы и т.д. Они могут также использоваться в сочетании с другими

системами безопасности, такими как системы видеонаблюдения и пожарной сигнализации.

Актуальность данного исследования связана с потребностью в улучшении системы контроля и доступа (СКУД) АО «АВТОВАЗ», что приведет к повышению эффективности функционирования предприятия и снижению затрат на оплату труда. Это достигается более точным учетом рабочего времени.

Цель данной ВКР "Разработка системы учета рабочего времени сотрудника АО «ВАЗСИСТЕМ» на основе событий системы контроля и управления доступом" заключается в создании эффективной и надежной системы учета рабочего времени, которая будет основываться на событиях, собираемых и анализируемых системой контроля и управления доступом).

Целью данной выпускной квалификационной работы является модернизация архитектуры, существующей распределенной информационной системы путем внедрения микросервисных технологий на основе реактивной архитектуры, что послужит основой для последующего полного перехода к микросервисной архитектуре.

Для достижения поставленной цели необходимо выполнить промежуточные задачи:

- изучение требований и анализ бизнес-процессов: провести детальное исследование требований и бизнес-процессов АО «ВАЗСИСТЕМ» в отношении учета рабочего времени сотрудников;
- проектирование системы учета рабочего времени: разработать архитектуру и дизайн системы учета рабочего времени, которая будет основываться на событиях системы контроля и управления доступом [12];
- анализ событий СКУД: изучить события, регистрируемые системой контроля и управления доступом;

- разработка программного обеспечения: на основе проектной документации разработать программное обеспечение для системы учета рабочего времени. Реализовать функциональность, связанную с событиями СКУД;
- тестирование и отладка: провести тестирование разработанной системы, включая проверку функциональности, корректности обработки событий и соответствие требованиям.

Объект исследования – система контроля и управления доступом АО «АВТОВАЗ».

Предметом исследования является СКУД, позволяющая осуществлять контроль над предоставлением доступа на вход и на выход в АО «АВТОВАЗ».

Методы исследования – микросервисные архитектуры.

Первая глава содержит описание СКУД в АО «ВАЗСИСТЕМ», характеристику текущего архитектурного решения и анализ работы системы контроля и управления доступом.

Вторая глава посвящена анализу программной реализации системы контроля и управления доступом, а также проектированию решения проблемы с точностью и достоверностью данных.

В третьей главе описаны инструменты разработки и процесс разработки программного модуля, тестирование работоспособности.

Результаты выполнения данной выпускной квалификационной работы имеют практический интерес и могут быть рекомендованы разработчикам высоконагруженных информационных систем.

1 Характеристика организации и экосистемы сервисов

1.1 Характеристика организации

АО «АвтоВАЗ» – советская и российская автомобилестроительная компания. Крупнейший производитель легковых автомобилей в Российской Федерации. Завод основан в 1966 году в городе Тольятти, в котором находятся штаб-квартира и основное производство. Входит в перечень системообразующих организаций России. Производит автомобили под собственной торговой маркой Lada, ранее выпускал автомобили марки Nissan, Renault, Datsun.

Компания является одним из ключевых игроков на рынке автомобильной промышленности в России, и ее производственные мощности позволяют выпускать около 500 тысяч автомобилей в год. АвтоВАЗ также активно работает над модернизацией своих производственных линий и внедрением новых технологий, чтобы повысить качество своей продукции и улучшить конкурентоспособность на рынке.

Дочерняя компания АвтоВАЗа АО «ВАЗСИСТЕМ» стала центром компетенций по внедрению и сопровождению программных продуктов на платформе 1С, на базе которых реализуются масштабные IT-проекты. Также она обеспечивает сопровождение и развитие многих других информационных систем, которые используются на заводе и предприятиях группы АВТОВАЗ.

Все началось в 1991 году, когда «АВТОВАЗ» и итальянская фирма «Logosystem» решили основать совместное предприятие для решения IT-задач.

У компании коммерческое название «INSOFT LADA» и более 160 программистов в штате, что позволяет назвать данную компанию одной

из крупнейших IT-компаний Самарской области по численности IT-персонала.

На АО «АвтоВАЗ» функционирует более 1300 систем. Среди них:

- система контроля и управления доступом;
- отдел безопасности информационных систем и технологий;
- отдел договорных отношений и расчётов;
- служба руководителя по инжинирингу;
- центр управления портфелем проектов и ресурсов;
- электронный документооборот учётных документов;
- информационная система безопасности.

АО «ВАЗСИСТЕМ» имеет государственную аккредитацию организации, осуществляющей деятельность в области информационных технологий.

1.2 Обзор существующих систем контроля и управления доступом

Существует множество систем контроля и управления доступом на рынке, каждая из которых имеет свои особенности и возможности. Рассмотрим несколько наиболее распространенных систем:

Honeywell Pro-Watch - это система управления доступом, которая предлагает широкий спектр функций, включая управление пропусками, мониторинг событий и систему видеонаблюдения. Она может работать на нескольких устройствах, включая считыватели биометрических данных и беспроводные устройства.

Lenel OnGuard - это программное обеспечение для систем управления доступом, которое обеспечивает максимальную защиту для зданий и сетей. Она включает в себя функции, такие как управление

доступом, видеонаблюдение и мониторинг. Она также может быть настроена для работы с различными устройствами и технологиями, включая биометрические считыватели.

Genetec Security Center - это система управления безопасностью, которая включает в себя функции управления доступом, видеонаблюдение и мониторинг. Она может работать на разных устройствах, включая камеры видеонаблюдения, считыватели идентификационных карт и биометрические сенсоры.

Bosch Access Engine - это программа управления доступом, которая предлагает широкий спектр функций, включая управление пропусками, мониторинг событий и систему видеонаблюдения. Она может работать на нескольких устройствах, включая считыватели биометрических данных и беспроводные устройства.

Avigilon Control Center - это система управления доступом, которая предлагает функции управления пропусками и мониторинга. Она также может быть настроена для работы с различными устройствами, включая считыватели идентификационных карт и биометрические сенсоры.

СКУД "Рубеж" - это российская система контроля и управления доступом. Она предназначена для обеспечения безопасности объектов любой сложности - от небольших офисов до крупных инфраструктурных объектов, таких как аэропорты, стадионы, торговые центры и т.д.

СКУД "Рубеж" обладает широкими возможностями настройки, что позволяет ее использовать на объектах любой сложности и размера. Система также имеет возможность интеграции с другими системами безопасности, такими как системы видеонаблюдения и охранной сигнализации, что обеспечивает еще более высокий уровень безопасности объектов.

Одним из распространённых типов СКУД является биометрическая система контроля доступа. Биометрическая система контроля доступа (БСКД) использует уникальные физиологические или

поведенческие характеристики человека для идентификации и аутентификации. Эта система использует физические характеристики человека, такие как отпечатки пальцев, радужка глаза или голос, для идентификации и контроля доступа. Эта система является более безопасной, чем система с картами доступа, поскольку биометрические характеристики невозможно украсть или потерять, как это может произойти с картами доступа.

Еще одним типом СКУД является система контроля доступа на основе сети. Она позволяет управлять доступом к объектам из любой точки с помощью сети Интернет. Эта система особенно полезна для компаний, которые имеют несколько офисов в разных городах или странах.

Системы контроля и управления доступом также могут быть интегрированы с другими системами безопасности, такими как системы видеонаблюдения или пожарной сигнализации. Такая интеграция позволяет создать комплексную систему безопасности, которая обеспечивает полную защиту объекта. Такая комплексная система безопасности с интегрированными системами контроля и управления доступом, видеонаблюдения и пожарной сигнализации позволяет сократить время реакции на происшествия, улучшить координацию мер безопасности и обеспечить более эффективную защиту объекта.

Одной из наиболее распространенных систем является электронная система контроля доступа. Электронная система контроля доступа (ЭСКД) представляет собой комплекс технических средств и программного обеспечения, разработанных для обеспечения безопасности и контроля доступа на объектах. В её состав входят такие компоненты, как карты доступа, считыватели, электронные замки и программное обеспечение для управления системой. С помощью карт доступа, которые носят сотрудники, система может контролировать доступ к зданию, офису или отдельным помещениям. Именно такая

система функционирует на АО «АвтоВАЗ». Схема, иллюстрирующая работу электронной системы контроля доступа, представлена на рисунке 1.

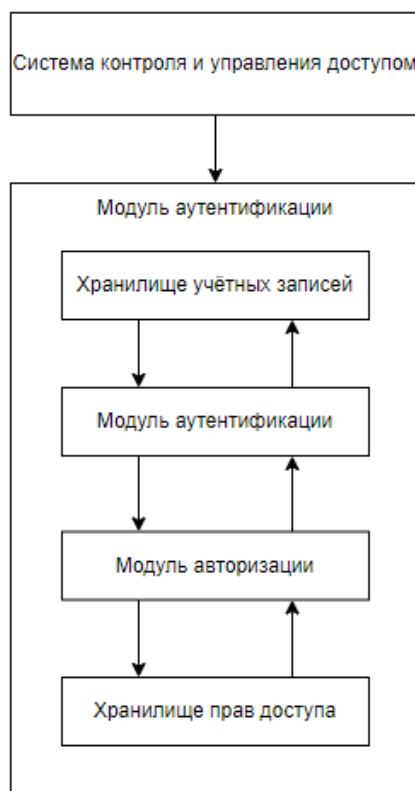


Рисунок 1 - Схема работы электронной системы контроля доступа

В этой схеме представлена система контроля и управления доступом, состоящая из нескольких модулей.

Первый модуль - модуль аутентификации, отвечает за проверку подлинности пользователя и проверку его учетных данных. Для этого используется хранилище учетных записей, где хранятся данные о всех пользователях системы. Если пользователь проходит проверку, то он получает доступ к следующему модулю - модулю авторизации.

Модуль авторизации, в свою очередь, проверяет права доступа пользователя и определяет, какие действия ему разрешено выполнить в системе (например, войти на территорию объекта). Для этого модуль

использует хранилище прав доступа, где хранятся данные о том, какие пользователи имеют доступ к каким ресурсам системы.

В целом, системы контроля и управления доступом являются важной частью систем безопасности для многих предприятий и организаций. Они обеспечивают безопасность и защиту сотрудников, посетителей и имущества, а также позволяют контролировать и отслеживать доступ к различным помещениям и объектам.

1.3 Анализ системы контроля и управления доступом на АО «ВАЗСИСТЕМ»

Для того, чтобы понять, как работает СКУД, реализованная компанией «ВАЗСИСТЕМ», на предприятии АО «АвтоВАЗ», нужно рассмотреть общую схему его работы. На рисунке 2 проиллюстрирована схема работы СКУД.

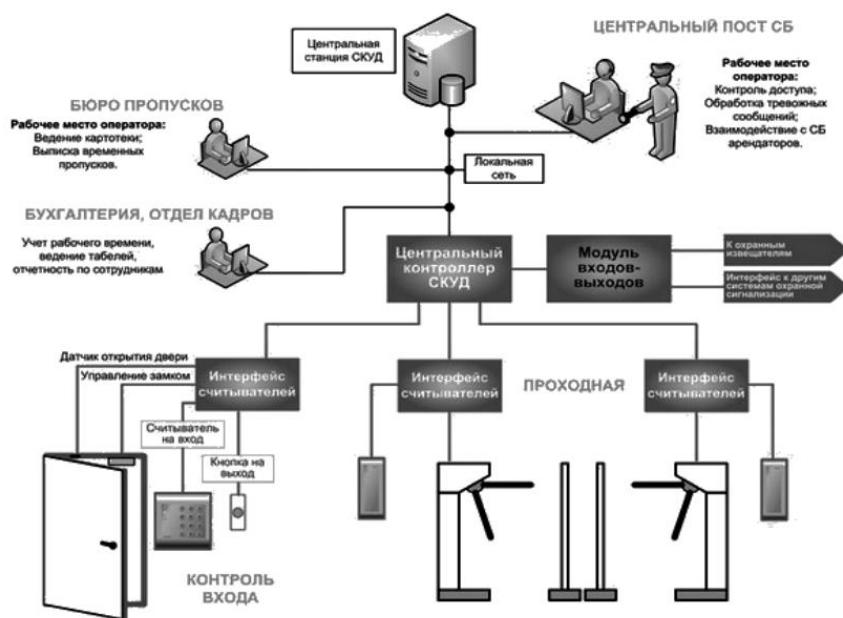


Рисунок 2 – схема работы СКУД

На АО «АвтоВАЗ» в качестве системы контроля и управления доступом используется «Рубеж».

СКУД "Рубеж" разработана компанией "Айпротек". С помощью "Рубежа" можно создавать электронные пропуска, определять права доступа для разных категорий пользователей, вести учет рабочего времени и многое другое.

В ПО СКУД «Рубеж» используется микросервисная архитектура. Такой подход к проектированию и разработке ПО, в котором приложение разбивается на мелкие и независимые компоненты, называемые микросервисами. Каждый микросервис отвечает за выполнение определенной функциональности и взаимодействует с другими микросервисами через API. Схема микросервисной архитектуры представлена на рисунке 3.

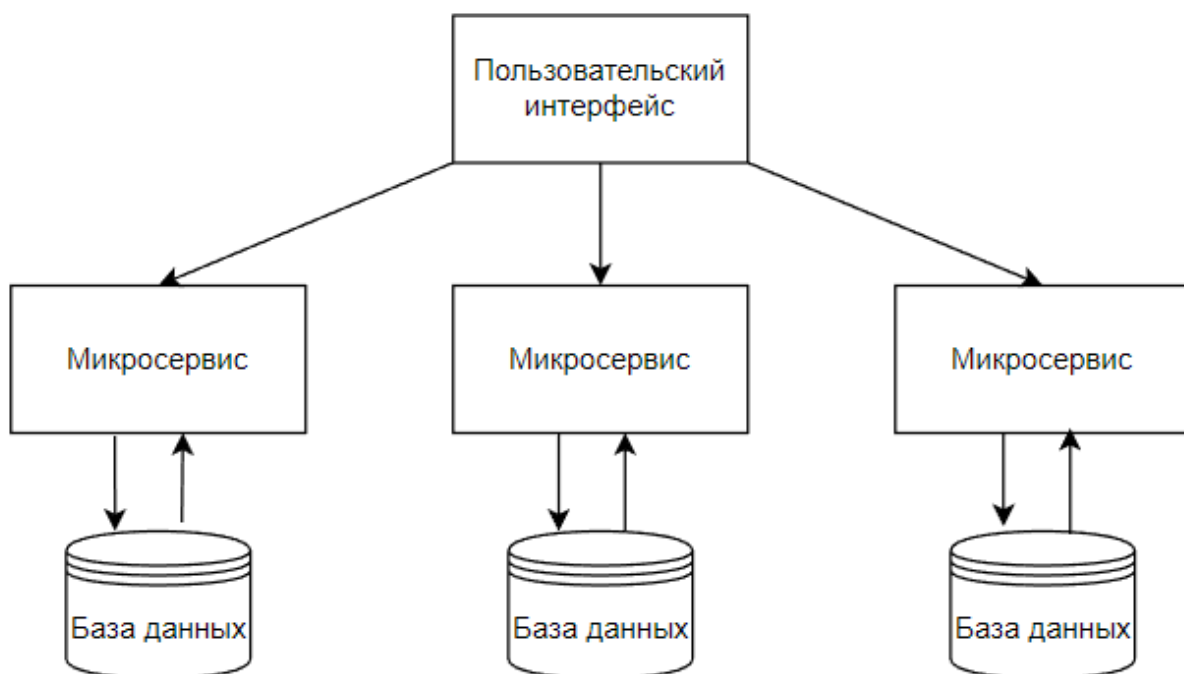


Рисунок 3 – схема микросервисной архитектуры

На данной схеме каждый микросервис отвечает за определенную функциональность: работу с базой данных, обработку запросов. Данное решение позволяет создавать гибкие и масштабируемые приложения.

Рассмотрим каждый уровень с рисунка 3. Уровень пользовательского интерфейса включает в себя веб-интерфейс, веб-сервер, автоматизированные системы управления бизнес-процессами. Уровень бизнес-логики включает в себя сервисы аутентификации и авторизации, сервисы управления правами доступа, сессиями, данными, бизнес-процессами, ресурсами, безопасностью. Уровень хранения данных включает в себя подуровень системы управления базами данных, реляционную СУБД, документоориентированную СУБД [2].

СКУД «Рубеж» состоит из нескольких компонентов: контроллеров доступа, считывателей карт, программного обеспечения для управления системой и базы данных для хранения информации о пользователях и их доступе. На АО «АвтоВАЗ» контроллеры устанавливаются на турникетах на контрольно-пропускных пунктах, а считыватели карт позволяют пользователю авторизоваться в системе, приложив к ним свою электронную карту.

В качестве контроллера на предприятии используется сетевой Web-контроллер STR20-IP-Ent на базе операционной системы Linux. Контроллер является главным устройством, к которому подключаются модули доступа и считыватели OSDP по интерфейсу RS-485, использующем протокол OSDP. Данный протокол используется для связи между контроллерами систем контроля доступа и устройствами считывания (считыватели карт). Каждый контроллер оснащен встроенным программным обеспечением, позволяющим настраивать систему, хранить базу данных, параметры доступа, а также журнал событий.

Контроллеры обеспечивают следующие функции:

- хранение и управление информацией о RFID-картах;
- хранение и управление информацией о персонале;
- хранение информации о режимах доступа, включая временные зоны;
- хранение информации о событиях;
- сохранение работоспособности при отсутствии связи с верхним уровнем программного обеспечения;
- индикацию режимов работы;
- питание внешних устройств, подключаемых к контроллерам;
- автоматическое управление проходами через турникеты;
- объединение контроллеров в кластер по сети Ethernet с совместным поведением;
- возможность использования функции глобального зонального контроля в пределах одного сегмента контроллеров.

В рассматриваемом web-контроллере реализовано встроенное сетевое ПО. Оно позволяет объединять контроллеры в кластер, создавая распределённую систему контроля доступа без использования выделенного сервера. Единый Web-интерфейс контроллеров используется для настройки и администрирования всей системы, а также для управления точками доступа [21].

Система контроля и управления доступом, которая реализована на АО «АвтоВАЗ», начинается с информационной системы «Охрана» с широким функционалом, который включает в себя регистрацию, выдачу пропусков сотрудникам. Данные лежат в «БД ORACLE 1». Также есть отдельная «БД ORACLE 2», где хранятся действующие пропуска сотрудников, а данные обновляются в режиме онлайн. И отдельным образом работает ПО СКУД с рассмотренными базами данных и контроллерами, которые установлены на турникетах в контрольно-пропускных пунктах. Схема работы системы

контроля и управления доступом, реализованной на АО «АвтоВАЗ», представлена на рисунке 4.

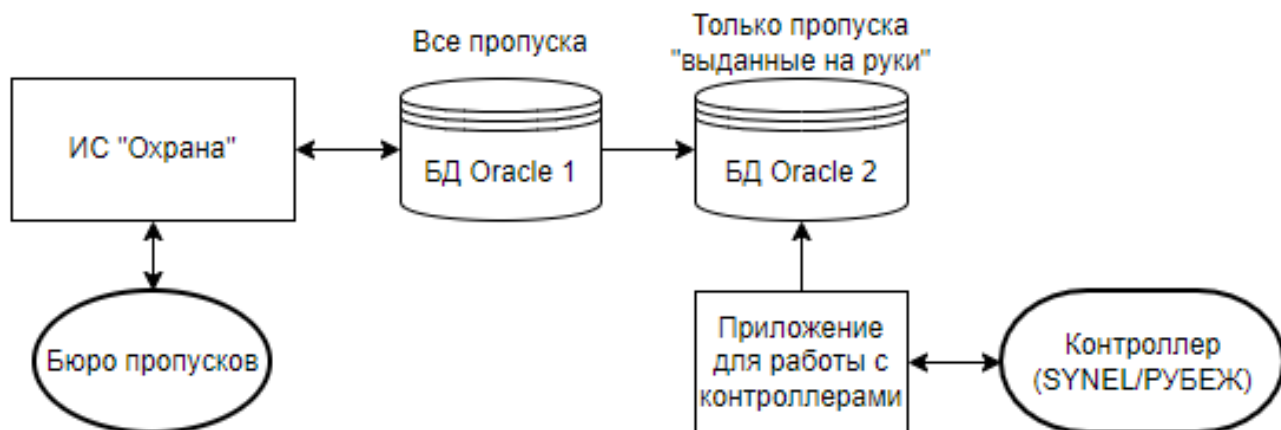


Рисунок 4 – схема работы системы контроля и управления доступом, реализованной на АО «АвтоВАЗ»

Система контроля и управления доступом, реализованная на АО «АвтоВАЗ», имеет многоуровневую архитектуру. Ее целью является разделение функциональности и ответственности между различными уровнями, что обеспечивает более гибкую, модульную и масштабируемую систему.

На предприятии АО «АвтоВАЗ» точкой доступа является турникет. Точка доступа турникет - это устройство, которое используется в системах контроля доступа для регулирования перемещения людей через определенную точку, например, вход или выход из здания, помещения или ограниченной зоны. Турникеты представляют собой поворотные или сдвижные барьеры, которые можно пройти только при наличии действительного разрешения доступа. Особенности такой точки доступа является физическое ограничение прохода и интеграция с другими системами безопасности (системы видеонаблюдения или системы

пожарной сигнализации). Рисунок 5 демонстрирует схему работы такой точки доступа.

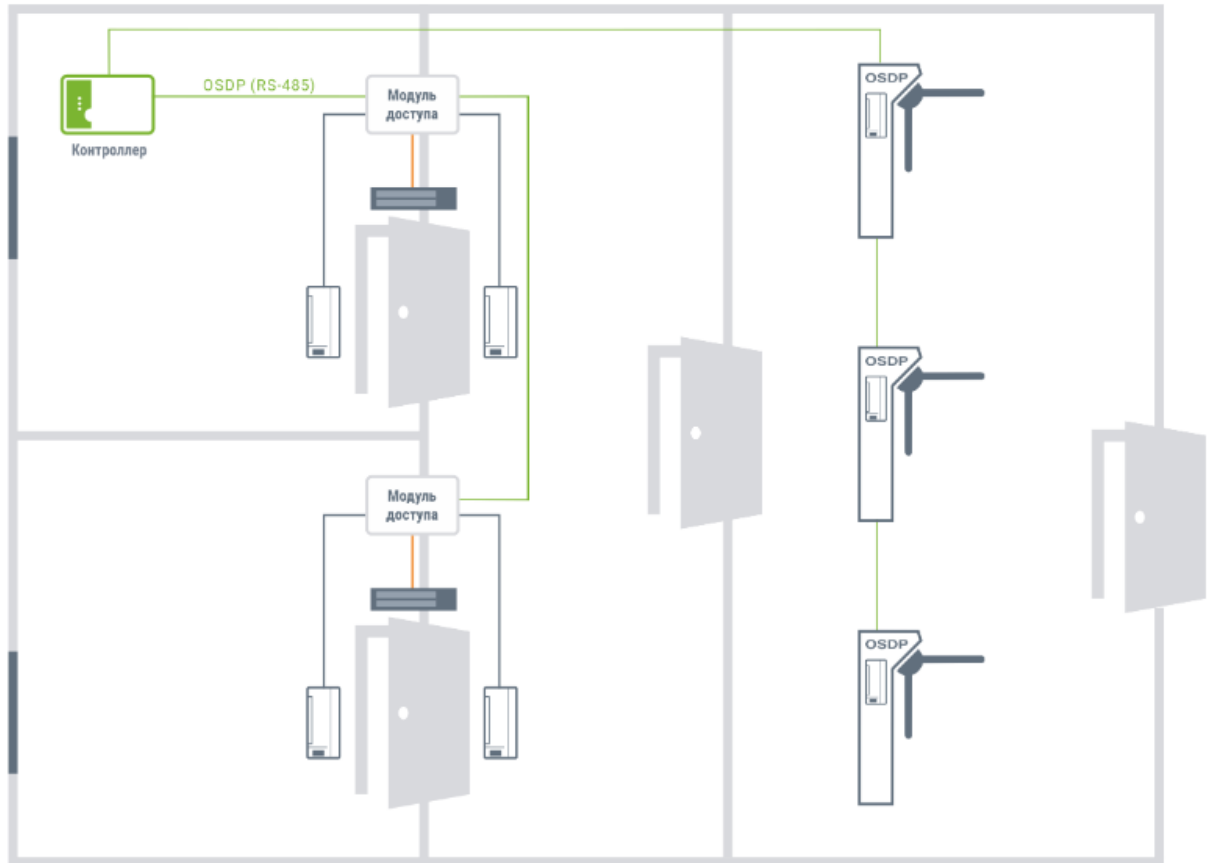


Рисунок 5 – точка доступа «Турникет»

Таким образом, на предприятии функционирует система контроля и управления доступом, позволяющая учитывать рабочее время сотрудников АО «АвтоВАЗ», а также позволяющая организовать работу контрольно-пропускных пунктов в тесной связи с системой безопасности.

2 Проектирование системы контроля и управления доступом

2.1 Анализ программной реализации СКУД на предприятии

Прежде чем перейти к анализу программной реализации СКУД на АО «АвтоВАЗ», рассмотрим блок-схему работы СКУД. Она представлена на рисунке 6.

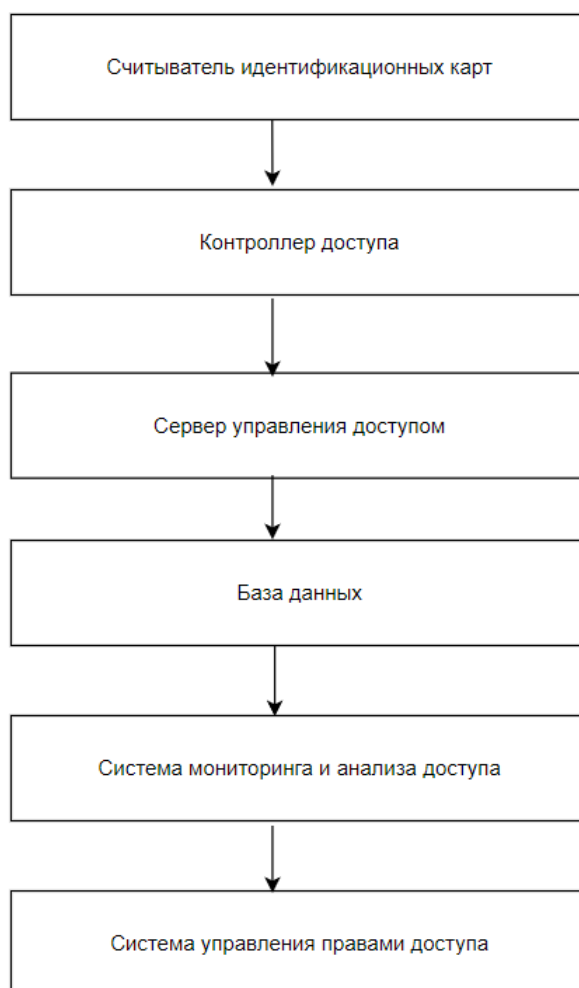


Рисунок 6 – блок-схема работы СКУД на АО «АвтоВАЗ»

Компоненты СКУД на рисунке 1 работают следующим образом: считыватели идентификационных карт считывают данные с карточек

сотрудников, проходящих через точки доступа. Полученные данные передаются на контроллеры доступа, которые контролируют доступ в соответствующие зоны. Контроллеры доступа обмениваются данными с серверами управления доступом, которые управляют правами доступа сотрудников в зависимости от их должностей и разрешений [5]. Данные о доступе и проходах сохраняются в базах данных для последующего использования в системах мониторинга и анализа доступа [15]. Системы мониторинга и анализа доступа используют данные из баз данных для мониторинга активности сотрудников, выявления неправильных действий и устранения уязвимостей в системе [20]. Системы управления правами доступа используют данные из баз данных и серверов управления доступом для управления правами доступа сотрудников, их должностями и разрешениями [3]-[4].

Таким образом, блок-схема состоит из шести компонентов, которые взаимодействуют друг с другом для обеспечения функционирования СКУД.

У каждого сотрудника имеются уникальные поля, позволяющие его идентифицировать в системе. Рисунок 7 иллюстрирует структуру данных запроса на создание сотрудника.

```
// Структура данных запроса на создание сотрудника
interface AddStaffRequest {
    uuid: string // ID
    name: string // Имя
    surname: string // Фамилия
    second_name: string // Отчество
    access_profile: string // ID профиля доступа
    access_point: string // ID личной точки доступа
    cards: string[] // Привязанные карты
    pin: string // Пин код
    biometry: {
        models: [] // Привязанная биометрия
    }
    department: string // Подразделение
    city: string // Город
    position: string // Должность
    car: string // Марка автомобиля
    birthday: string // День рождения
    age: number // Возраст
    car_number: string // Номер машины
    banned: boolean // Признак блокировки
}
```

Рисунок 7 – структура данных запроса на создание сотрудника

Например, каждый сотрудник имеет уникальный ID профиля доступа и уникальный ID личной точки доступа, номер автомобиля, а также привязанные карты, позволяющие осуществлять проход через контрольно-пропускные пункты. Также у некоторых сотрудников может быть привязанная биометрия [11]-[13].

Также у контроллеров есть web-интерфейс. В РУБЕЖ своё встроенное программное обеспечение в контроллер, веб-интерфейс, где настраивается (через веб-браузер) сам контроллер РУБЕЖ. Рисунок 8 показывает, как выглядит заполнение полей о сотруднике в web-интерфейсе контроллера СКУД «Рубеж».

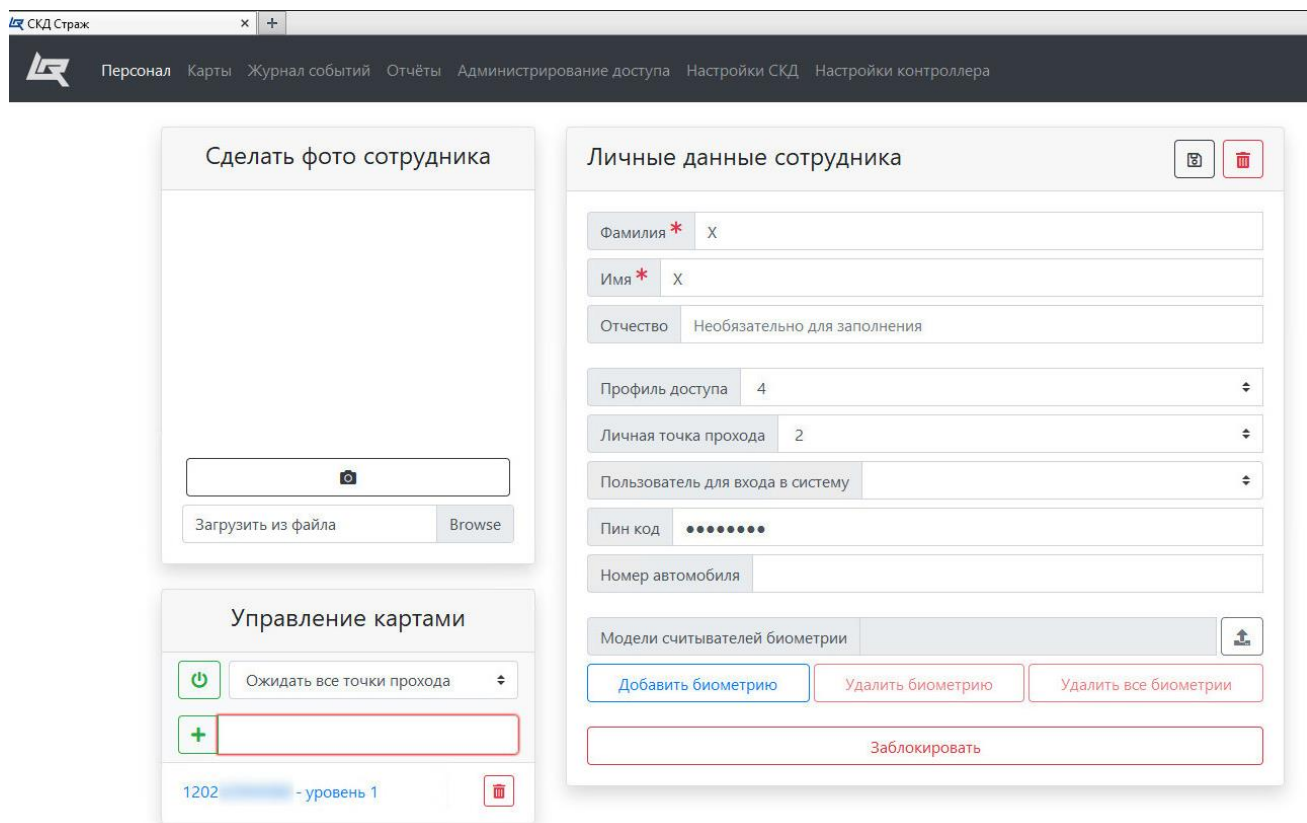


Рисунок 8 - заполнение полей о сотруднике в web-интерфейсе контроллера СКУД «Рубеж»

У каждой карты имеются уникальные поля, позволяющие её идентифицировать в системе. Рисунок 9 иллюстрирует структуру данных запроса на добавление карты в систему

```
// Структура данных запроса на добавление карты в систему
interface AddCardRequest {
  code: string // Номер карты
  privilege: number // Уровень привилегий
  disposable: boolean // Временная карта
  expire: number // Действует до (timestamp)
  comment: string // Произвольное описание
  user_uuid: string // Привязка к сотруднику по uuid
  ext_privileges: string[] // Флаги дополнительных привилегий
}
```

Рисунок 9 – структура данных на добавление карты в систему

Карта может быть временной и постоянной. Каждая карта имеет свой уровень привилегий, позволяющие иметь доступ к различным КПП.

Рассмотрим фрагмент кода, заполняющий структуру данных о сотруднике и о карте в системе. Данный код иллюстрирует рисунок 10.

```
// Добавим карту
await AddCard({
  code: "514515516",
  privilege: 1,
  disposable: false,
  expire: 1790629199999,
  comment: "Добавлена super 28.09.2021",
  user_uuid: "",
  ext_privileges: [
    "SET_AP_STATE",
    "SET_ON_GUARD",
    "UNSET_ON_GUARD",
    "START_AUTOMATION",
    "CONFIRM_ACCESS",
  ],
}).catch(e => {
  throw new Error(e.message)
})

// Добавим сотрудника и выдадим ему нашу карту
await AddStaff({
  uuid: "8395dc90-204d-11ec-bdf5-47449b59fb8a",
  second_name: "Семенов",
  surname: "Тестов",
  access_profile: "cb00b91-8f68-11ea-bd04-0dd3c24f5be3",
  access_point: "1fbe9300-8799-11e9-9d26-a9d9404ec35e",
  cards: ["514515516"],
  pin: "1415",
  biometry: {
    models: [],
  },
  department: "Департамент разработки",
  city: "Абакан",
  position: "Кочегар",
  name: "Иван",
  car: "Kia",
  birthday: "21.06.1984",
  age: 37,
  car_number: "y555yy",
  banned: false,
}).catch(e => {
  throw new Error(e.message)
})

// Если всё прошло хорошо – сотрудник добавлен, карта привязана
```

Рисунок 10 – программный код, с помощью которого заполняется структура данных о сотруднике и о карте в системе.

Далее рассмотрим функцию добавления сотрудника в систему. На рисунке 11 представлен программный код, позволяющий добавлять нового сотрудника в систему.

```
export function AddStaff(data: AddStaffRequest) {
  return new Promise((resolve, reject) => {
    // Преобразуем данные в строку для передачи в теле запроса
    let bodyParams = JSON.stringify(data);

    // Параметры запроса
    const options = {
      hostname: global.config.hostname, // Адрес контроллера
      port: global.config.port, // Порт
      path: '/api/v1/staff', // Путь API метода
      method: 'POST', // Метод запроса
      headers: {
        'Content-Type': 'application/json',
        'Content-Length': Buffer.byteLength(bodyParams),
        'Authorization': `Bearer ${global.token}` // Здесь добавляем заголовок авторизации с ранее полученным токеном
      },
    };

    // Запрос к серверу
    const req = https.request(options, (response) => {
      let data = ''
      // Обработчик данных с сервера
      response.on('data', (chunk) => {
        data += chunk;
      });
      // Обработчик завершения запроса
      response.on('end', () => {
        // Успешно
        if (response.statusCode === 201) {
          // Декодируем и резолвим полученные данные
          resolve(JSON.parse(data));
        }
        // С ошибкой
        else reject(JSON.parse(data));
      })
    });

    // Отправка запроса
    req.write(bodyParams);

    // Ошибка запроса
    req.on('error', (error) => {
      throw error
    });

    req.end();
  })
}
```

Рисунок 11 – функция, реализующая добавление сотрудника в систему

Функция AddStaff отправляет POST запрос к API контроллера для добавления нового сотрудника в систему. Эта функция принимает объект data типа AddStaffRequest в качестве аргумента. Этот объект содержит информацию о новом сотруднике, которого необходимо добавить в систему.

Таким образом, данная функция выполняет асинхронный POST запрос к API контроллера с информацией о новом сотруднике, а затем принимает запрос на его добавление или отклоняет его.

Рассмотрим функцию авторизации пользователя в системе. Функцию авторизации пользователя в системе иллюстрирует рисунок 12.

```
export function Auth(login, password): Promise<AuthResponseSuccess> {
  return new Promise((resolve, reject) => {
    // Создаем объект с данными для авторизации
    const authData: AuthRequest = {
      login, password
    }

    // Преобразуем в строку для передачи в теле запроса
    let bodyParams = JSON.stringify(authData);

    // Параметры запроса
    const options = {
      hostname: global.config.hostname, // Адрес контроллера
      port: global.config.port, // Порт
      path: '/api/v1/login', // Путь API метода
      method: 'POST', // Метод запроса
      headers: {
        'Content-Type': 'application/json',
        'Content-Length': Buffer.byteLength(bodyParams),
      },
    };

    // Запрос к серверу
    const req = https.request(options, (response) => {
      let data = ''
      // Обработчик данных с сервера
      response.on('data', (chunk) => {
        data += chunk;
      });
      // Обработчик завершения запроса
      response.on('end', () => {
        // Успешно
        if (response.statusCode === 200) {
          // Декодируем и резолвим полученные данные
          resolve(JSON.parse(data));
        }
        // С ошибкой
        else reject(JSON.parse(data));
      })
    });

    // Отправка запроса
    req.write(bodyParams);

    // Ошибка запроса
    req.on('error', (error) => {
      throw error
    });

    req.end();
  })
}
```

Рисунок 12 – функция авторизации пользователя в системе

Функция принимает два аргумента - логин и пароль пользователя. Далее создается объект с переданными данными, который преобразуется в строку. Эта строка будет использоваться в теле запроса. Затем задаются параметры запроса в объекте `options`. Этот объект содержит информацию о том, куда и какой запрос нужно отправить – адрес сервера, порт, путь до API-метода, метод запроса, заголовки запроса. Далее создается запрос с помощью функции `httpsrequest`, которая принимает параметры запроса и функцию-обработчик, которая будет вызвана после получения ответа от сервера. В этой функции-обработчике первым делом создается переменная, в которую будут добавляться данные, полученные от сервера. Затем устанавливается обработчик события на объекте. В данном случае данные собираются в одну строку. Далее устанавливается обработчик события `end` на объекте `response`. Он вызывается после того, как все данные были получены. В данном случае функция проверяет статус-код ответа - если он равен 200, то данные декодируются из строки JSON с помощью функции `JSON.parse` и передаются в функцию `resolve` в виде объекта типа `AuthResponseSuccess`. Если статус-код не равен 200, то данные декодируются и передаются в функцию `reject` в виде объекта типа `AuthResponseError`.

В рассматриваемой функции отсутствует система контроля качества данных, что может повлечь за собой снижение эффективности системы: неправильно учтенное время прихода и ухода может привести к тому, что сотрудникам будет неправильно начислено рабочее время.

2.2 Проектирование решения проблемы с точностью и достоверностью данных

Для решения проблемы с точностью и достоверностью данных потребуется разработать систему контроля качества данных. СККД -

набор процессов, инструментов и методов, которые используются для обеспечения высокого качества данных, связанных с управлением доступом на объекте или в организации.

В целом, система контроля качества данных является важным компонентом СКУД, который обеспечивает высокое качество данных и повышает эффективность работы системы [16]. С помощью правильной системы контроля качества данных можно увеличить надежность, безопасность и производительность СКУД, что имеет важное значение для эффективной работы бизнеса и защиты критически важных ресурсов.

Один из ключевых процессов СККД в СКУД - это контроль качества данных на этапе ввода. На этом этапе данные проверяются на соответствие заранее заданным правилам и ограничениям, таким как форматы, типы данных, длина полей и т.д. В случае нарушения правил ввода данных система должна предупредить пользователя и предложить исправить ошибки [14].

Другим важным аспектом СККД в СКУД является контроль качества данных в процессе работы системы. На этом этапе данные могут быть проверены на дублирование, неправильное форматирование, неполные данные и т.д. В случае обнаружения ошибок СКУД может предложить исправление данных или автоматически исправить их при наличии соответствующих правил [17]-[18].

К тому же, качественные данные позволяют системе контроля доступа функционировать более эффективно и улучшают производительность организации. Если данные актуальны и соответствуют текущей ситуации, система может быстро принимать решения о доступе и обрабатывать запросы сотрудников без задержек или ошибок. Это также помогает предотвратить необходимость вручную вмешиваться и корректировать данные, что экономит время и ресурсы.

Разрабатываемое решение будет содержать в себе 6 функций, каждая из которых позволит произвести контроль качества данных. В таблице 1 представлены функции и их роль в предлагаемом решении.

Таблица 1 – Функции СККД

Название функции	Роль функции
checkData	Проверка правил ввода данных
check_duplicates	Проверка данных на наличие дубликатов
check_completeness	Проверка полноты данных
check_format	Проверка правильности формата данных
check_data_integrity	Проверка целостности данных
check_data_quality	Проверка качества данных

Разработка данных функций позволит реализовать систему контроля качества данных, что повлечет за собой увеличение точности учёта рабочего времени сотрудников АО «АвтоВАЗ».

Внедрение системы контроля качества данных может значительно улучшить работу СКУД, поскольку позволяет выявлять и исправлять ошибки в данных, связанные с отсутствием или неправильным заполнением информации, а также с плохим качеством оборудования и его неисправностью. Это помогает уменьшить количество ложных срабатываний, повысить точность и достоверность данных, а также

повысить эффективность работы СКУД в целом. Кроме того, система контроля качества данных позволяет производить анализ производительности и эффективности СКУД.

2.3 Проектирование решения проблемы учёта времени на перерывы при расчёте рабочего времени сотрудника в СКУД

Проблема отсутствия учета времени на перерывы в СКУД заключается в том, что система не учитывает или не фиксирует временные интервалы, когда сотрудник покидает рабочее место для перерыва, например, на обед или отдых.

Это означает, что в СКУД нет механизма, позволяющего корректно учитывать и управлять временем, проведённым сотрудником на перерыве. Как следствие, система не сохраняет информацию о начале и окончании перерывов, не учитывает их продолжительность и не вносит соответствующие изменения в данные о рабочем времени сотрудника [6].

Для наглядности будет необходимо рассмотреть статистику. Согласно статистике, 21,9% жителей России курят. В среднем рабочий тратит на курение 40 минут. Рассмотрим столбчатую диаграмму, показывающую, какое количество времени провёл человек на работе. Она показана на рисунке 13.

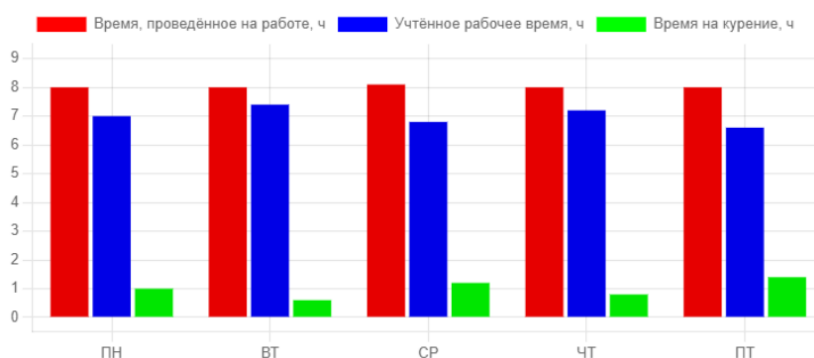


Рисунок 13 – диаграмма времени человека на работе

По данной диаграмме можно сделать вывод, что человек провёл на работе 8 часов (а в среду – 8 часов, 6 минут), при этом СКУД засчитает только от 6,6 до 7,4 часов. Однако рабочий проводил 8 часов на предприятии, что и должно быть учтено. Также переработка в среду (равная 6 минутам) должна быть зафиксирована, ведь в случае опоздания данного человека (или в случае, если он уйдёт раньше с работы), нужно учитывать все его переработки, чтобы не было наложено неоправданных санкций.

Таким образом, отсутствие учета времени на перерывы может привести к несправедливому учёту рабочего времени, неправильному расчёту оплаты труда, нарушению рабочих режимов и расписаний, а также недостоверности данных о рабочем времени. Это может создать недовольство среди сотрудников, нарушить баланс между работой и отдыхом, а также затруднить мониторинг и управление рабочей активностью. Проблема учёта перерывов в работе влечёт за собой нарушение рабочего расписания, недостаточный отдых сотрудников, нарушение рабочего процесса, а также неточность данных и учётных записей [8].

Для решения данной проблемы необходимо разработать и внедрить систему, которая будет учитывать и фиксировать время на перерывы сотрудников. Это может быть реализовано путем интеграции СКУД с другими системами, такими как система учета рабочего времени или система управления персоналом. Такая интеграция позволит точно отслеживать время на перерывы и включать его в общий учёт рабочего времени сотрудника [19].

Разрабатываемое решение и будет содержать в себе 3 функции, которые позволят исправить проблему отсутствия учета времени на

перерывы. В таблице 2 представлены функции и их роль в предлагаемом решении.

Таблица 2 – Функции, позволяющие исправить проблему отсутствия учета времени на перерывы

Название функции	Роль функции
updateWorkingHours	Обновляет рабочие часы сотрудника в зависимости от типа события
getEmployeeData	Получение данных о сотруднике на указанную дату
saveEmployeeData	Сохранение данных о сотруднике на указанную дату

Разработка данных функций позволит реализовать алгоритм, позволяющий решить проблему отсутствия учёта времени на перерывы, что повлечет за собой увеличение точности учёта рабочего времени сотрудников АО «АвтоВАЗ».

Внедрение данных функций может значительно улучшить работу СКУД, так как исправление проблемы учёта времени на перерывы при расчёте рабочего времени сотрудника позволит производить верный расчёт заработной платы, а также снизит риск наложения неоправданных санкций.

3 Реализация системы контроля и управления доступом

3.1 Выбор средств реализации

Для интеграции систем учета рабочего времени в СКУД использовались различные технологии, включая API-интерфейсы, протоколы синхронизации данных и технологии передачи данных в режиме реального времени. При выборе технологий для интеграции было учтено соответствие требованиям системы безопасности и защиты персональных данных [9]-[10].

Для решения проблемы точности и достоверности данных в СКУД были использованы технологии автоматического контроля качества данных, такие как проверка данных на соответствие формату и допустимым значениям, а также исключение дубликатов и пропущенных значений.

В выборе технологий были учтены требования к производительности, масштабируемости и надежности системы, а также наличие необходимой экспертизы и опыта разработки в выбранных технологиях.

Одно из условий выполнения данной выпускной квалификационной работы – реализация программ, которые решают проблемы актуальной СКУД. Так как СКУД на предприятии реализована на языке программирования TypeScript, программы будут реализованы на нём.

TypeScript – это язык программирования, разработанный Microsoft, который является надстройкой над JavaScript. TypeScript добавляет в JavaScript статическую типизацию и другие возможности, такие как

классы, интерфейсы, наследование и т.д. Это делает код более читаемым и поддерживаемым, особенно для больших проектов.

Важно отметить, что TypeScript компилируется в JavaScript, что позволяет использовать его на любых платформах, где поддерживается JavaScript, включая браузеры и серверные приложения. TypeScript также предоставляет мощную среду разработки, включая интегрированные средства отладки и инструменты для автоматического поиска ошибок [7].

TypeScript получил широкую поддержку в сообществе разработчиков и стал одним из самых популярных языков программирования. Он используется для разработки веб-приложений, мобильных приложений, серверных приложений и многих других типов программного обеспечения.

При выборе интегрированной среды разработки следует учитывать используемый язык программирования. Существует три наиболее популярных среды разработки с поддержкой TypeScript: Visual Studio Code, Eclipse и WebStorm.

Eclipse – это интегрированная среда разработки (IDE) для языка Java, но она также поддерживает TypeScript с помощью плагинов и расширений. Она имеет множество функций, таких как поддержка отладки, автодополнение, интеллектуальный анализ кода и многие другие, которые делают разработку на TypeScript более эффективной.

Visual Studio Code – это бесплатный редактор кода, который поддерживает множество языков программирования, включая TypeScript. Он имеет множество расширений и инструментов, которые позволяют эффективно разрабатывать и отлаживать TypeScript-приложения.

WebStorm – это интегрированная среда разработки, разработанная компанией JetBrains. Она также поддерживает различные языки программирования, в том числе TypeScript. В WebStorm есть функции проверки синтаксиса, автодополнение, отладка и улучшенная интеграция с системами контроля версий.

Поскольку существующие среды разработки имеют схожие функциональные возможности, необходимо провести сравнение на основе следующих критериев:

- удобство интерфейса;
- современность среды разработки;
- работа с плагинами и библиотеками – под работой с плагинами и библиотеками имеется ввиду удобство в их подключении;
- опыт использования – персональный опыт в использовании данной среды разработки.

Результат сравнения представлен в таблице 3.

Таблица 3 – Сравнение интегрированных сред разработки

Критерий выбора	Eclipse	WebStorm	Visual Studio Code
Удобство интерфейса	7	6	8
Современность среды разработки	7	6	8
Работа с плагинами и библиотеками	6	7	9
Опыт использования	5	0	9
Итого:	25	19	34

После сравнения трех интегрированных сред разработки, было принято решение использовать Visual Studio Code. Эта среда имеет современный и функциональный интерфейс, а также удобные инструменты для подключения плагинов и библиотек, в том числе для TypeScript. Одним из наиболее важных преимуществ является опыт разработки в данной среде.

В связи с тем, что существующая система использует PostgreSQL в качестве хранилища данных, будет использована эта СУБД для журнала событий.

PostgreSQL - это объектно-реляционная СУБД, которая основана на языке SQL. Ее основными особенностями являются высокая надежность, расширяемость и производительность.

3.2 Реализация системы контроля качества данных

Реализация системы контроля качества данных в СКУД позволяет улучшить ее работу и повысить качество данных, что в свою очередь способствует более эффективной работе системы учёта рабочего времени.

Перейдём к реализации функций, которые позволяют увеличить качество данных, поступающих в СКУД. На рисунках 14, 15, 16, 17, 18, 19 представлена реализация функций системы контроля качества данных.

```
function checkData(inputData) {  
    // Проверяем формат введенных данных  
    if (!inputData.match(/^[A-Za-z0-9_-]*$/)) {  
        return false;  
    }  
    // Проверяем длину поля  
    if (inputData.length > 20) {  
        return false;  
    }  
    // Проверяем тип данных  
    if (typeof inputData !== "string") {  
        return false;  
    }  
    return true;  
}
```

Рисунок 14 – реализация функции проверки ввода данных

```
function checkDuplicates(inputData) {  
  // Проверяем наличие дубликатов  
  if (inputData.length !== new Set(inputData).size) {  
    return true;  
  }  
  return false;  
}
```

Рисунок 15 – реализация функции проверки данных на наличие дубликатов

```
function checkCompleteness(inputData) {  
  // Проверяем полноту данных  
  if (Object.values(inputData).includes(null)) {  
    return false;  
  }  
  return true;  
}
```

Рисунок 16 – реализация функции проверки полноты данных

```
function checkFormat(inputData) {  
  // Проверяем правильность формата данных  
  if (!/^\d{4}-\d{2}-\d{2}$/.test(inputData.date)) {  
    return false;  
  }  
  if (!/^[A-Za-z\s]*$/.test(inputData.name)) {  
    return false;  
  }  
  return true;  
}
```

Рисунок 17 – реализация функции проверки правильности формата данных

```

function checkDataIntegrity(df) {
  // Проверяем целостность данных
  // Проверяем, что все записи имеют уникальный идентификатор
  if (!df.index.isUnique) {
    throw new Error("Data integrity error: duplicate index values found");
  }

  // Проверяем, что нет пропущенных значений
  if (df.isnull().values.some((value) => value)) {
    throw new Error("Data integrity error: missing values found");
  }
}

```

Рисунок 18 – реализация функции проверки целостности данных

```

function checkDataQuality(df) {
  // Проверяем качество данных
  // Проверяем, что значения в колонке "время" находятся в правильном формате
  if (!df['время'].dtype.name.startsWith('datetime64[')) {
    throw new Error("Data quality error: wrong format in column "время");
  }

  // Проверяем, что значения в колонке "место" находятся в списке допустимых значений
  const allowedPlaces = ['вход', 'выход'];
  if (!df['место'].isin(allowedPlaces).all()) {
    throw new Error("Data quality error: invalid values found in column "место");
  }
}

```

Рисунок 19 – реализация функции проверки качества данных

Функция на рисунке 14 использует регулярное выражение для проверки формата введенных данных, а также проверяет длину поля и тип данных. При несоответствии правил ввода функция возвращает значение False. Функция на рисунке 15 проверяет данные на наличие дубликатов. Функция на рисунке 16 проверяет полноту данных. Функция на рисунке 17 проверяет правильность формата данных. Функция на рисунке 18 проверяет целостность данных. Функция на рисунке 19 проверяет качество данных.

Общий подход к разработке программы для системы контроля качества данных в СКУД заключается в тщательном анализе данных, определении правил и требований к качеству данных и разработке соответствующих

функций и алгоритмов для автоматического контроля и улучшения качества данных в системе. Правильный контроль и управление качеством данных в СКУД является важным компонентом обеспечения безопасности и эффективности системы.

3.3 Реализация алгоритма решения проблемы учёта времени на перерывы

Реализация алгоритма решения проблемы учёта времени на перерывы позволит проводить точный расчет заработной платы и снизит вероятность наложения неоправданных санкций.

Перейдём к реализации функций для решения данной описанной проблемы. На рисунках 20, 21, 22 представлена реализация функций алгоритма решения проблемы учёта времени на перерывы.

```
function updateWorkingHours(eventType: string): void {
  const current_date: moment.Moment = moment().startOf('day');

  // Получение данных о рабочем времени сотрудника из базы данных СКУД
  const employeeData: EmployeeData | null = getEmployeeData(current_date);

  if (employeeData) {
    // Получение текущего статуса сотрудника (на работе или на перерыве)
    const currentStatus: string = employeeData.status;

    if (eventType === 'Выход на перерыв') {
      // Проверка текущего статуса сотрудника
      if (currentStatus === 'На работе') {
        // Обновление статуса сотрудника на "На перерыве"
        employeeData.status = 'На перерыве';
        employeeData.break_start_time = moment();
        saveEmployeeData(current_date, employeeData);
      }
    } else if (eventType === 'Возвращение с перерыва') {
      // Проверка текущего статуса сотрудника
      if (currentStatus === 'На перерыве') {
        // Расчет продолжительности перерыва
        const break_start_time: moment.Moment | null = employeeData.break_start_time;
        if (break_start_time) {
          const break_duration: moment.Duration = moment.duration(moment().diff(break_start_time));

          // Обновление статуса сотрудника на "На работе"
          employeeData.status = 'На работе';
          employeeData.total_break_duration.add(break_duration);
          saveEmployeeData(current_date, employeeData);
        }
      }
    }
  }
}
```

Рисунок 20 – реализация функции обновления рабочих часов сотрудника в зависимости от типа события

```

function getEmployeeData(date: moment.Moment): EmployeeData | null {
  // Получение данных о сотруднике из базы данных СКУД на указанную дату
  const employeeData: EmployeeData = {
    status: 'На работе',
    break_start_time: null,
    total_break_duration: moment.duration(0, 'minutes'),
  };

  return employeeData;
}

```

Рисунок 21 – реализация функции получения данных о
сотруднике на указанную дату

```

// Функция для сохранения данных сотрудника
function saveEmployeeData(employeeName: string, date: string): void {
  // Подключение к базе данных
  const conn: sqlite3.Database = new sqlite3.Database('skud_database.db');

  // Вставка данных о сотруднике в таблицу
  conn.run('INSERT INTO employees (name, date) VALUES (?, ?)', [employeeName, date], function (err) {
    if (err) {
      console.error('Ошибка при вставке данных о сотруднике:', err);
    } else {
      console.log('Данные о сотруднике успешно сохранены');
    }
  });

  // Закрытие соединения с базой данных
  conn.close();
}

```

Рисунок 22 – реализация функции сохранения данных о сотруднике на
указанную дату

Функция на рисунке 20 принимает тип события `event_type` и обновляет рабочие часы сотрудника в зависимости от типа события. Функция на рисунке 21 получает данные о сотруднике из базы данных СКУД на указанную дату. Функция на рисунке 22 принимает имя сотрудника и дату, устанавливает соединение с базой данных, вставляет данные о сотруднике в таблицу и сохраняет изменения в базе данных.

Правильный учёт рабочего времени (учитывая перерывы и переработки) позволит верно рассчитывать заработную плату и позволит видеть реальное количество времени, которое работал сотрудник. Реализованная программа является важным компонентом эффективности системы.

3.4 Тестирование системы

В данном разделе проводится описание процесса тестирования системы контроля и управления доступом (СКУД) в контексте разработки системы учета рабочего времени сотрудника АО «ВАЗСИСТЕМ» на основе событий СКУД.

Цели тестирования СКУД включают проверку различных аспектов функциональности системы. Они помогают гарантировать, что СКУД работает должным образом и соответствует требованиям, предъявляемым к системе учета рабочего времени сотрудника. Ниже приведены некоторые типичные цели тестирования СКУД:

1. Проверка правильности работы: Одной из основных целей тестирования является убедиться, что СКУД выполняет свои функции правильно. Это включает проверку точности определения времени прихода и ухода сотрудников, корректное управление доступом и регистрацию событий.
2. Обнаружение и исправление ошибок: Тестирование позволяет выявить ошибки и дефекты в работе СКУД. Это могут быть ошибки в логике системы, неправильное чтение данных считывателей карт, проблемы с соединением и другие. Выявленные ошибки затем исправляются разработчиками для обеспечения надлежащей работы системы.
3. Проверка безопасности: Тестирование СКУД включает проверку безопасности системы. Это включает проверку механизмов аутентификации и авторизации, защиту данных от

несанкционированного доступа, обнаружение и предотвращение возможных уязвимостей.

Результаты тестирования СКУД являются важной частью процесса разработки и позволяют оценить качество работы системы. В результате тестирования могут быть обнаружены различные проблемы, ошибки и недочеты, которые требуют исправления. На рисунках 23, 24, 25 приведены результаты тестирования СКУД.

1 Данные по проходам

Excel Печать Колонки Записей

Фильтр

ВРЕМЯ	НАПР.	КПП	ПРОХОД	ПРОВЕРКА	ПРОПУСК	ТИП	ФАМИЛИЯ	ИМЯ	ОТЧЕСТВО	КОД	ПОДРАЗДЕЛЕНИЕ	ПОСТ.№
16.05.2023 09:48:22	ВЫХОД	4	Да	ОК	25	КОМ				M04ZE	ЧОП Славяне-Сервис	
16.05.2023 09:47:08	ВХОД	4	Да	ОК	25	КОМ				M04ZE	ЧОП Славяне-Сервис	
16.05.2023 09:33:04	ВХОД	4	Да	ОК	24	ИП				92020	ОПЕРАТИВНЫЙ ОТДЕЛ	
16.05.2023 09:32:02	ВЫХОД	4	Да	ОК	24	ИП				92020	ОПЕРАТИВНЫЙ ОТДЕЛ	
16.05.2023 08:09:27	ВХОД	4	Да	ОК	25	КОМ				M04ZE	ЧОП Славяне-Сервис	
16.05.2023 04:05:15	ВХОД	4	Да	ОК	25	ВРМ				M04ZE	ЧОП Славяне-Сервис	
16.05.2023 02:38:45	ВХОД	4	Да	ОК	25	ВРМ				M04ZE	ЧОП Славяне-Сервис	
16.05.2023 01:36:51	ВХОД	4	Да	ОК	25	ВРМ				M04ZE	ЧОП Славяне-Сервис	

1 по 8 из 8

Рисунок 23 – данные по проходам в СКУД

СКУД Страж

Персонал Карты Журнал событий Отчёты Администрирование доступа Настройки СКУД Настройки контроллера

admin (10:00)

Авто обновление Доступ Аудит Оповещения с 16 мая 2023 г. 00:00 по 16 мая 2023 г. 23:59

#	Время	Событие	Объект действия	Кто совершил
8	16.05.2023, 09:48:22	Совершён проход по разрешению внешней системы	ТП "2" (2:1:1 -> 1:1:1)	Х Х (карта #193)
7	16.05.2023, 09:47:08	Совершён проход по разрешению внешней системы	ТП "2" (1:1:1 -> 2:1:1)	Х Х (карта #193)
6	16.05.2023, 09:33:04	Совершён проход по разрешению внешней системы	ТП "2" (1:1:1 -> 2:1:1)	Х Х (карта #176)
5	16.05.2023, 09:32:02	Совершён проход по разрешению внешней системы	ТП "2" (2:1:1 -> 1:1:1)	Х Х (карта #176)
4	16.05.2023, 08:09:27	Совершён проход по разрешению внешней системы	ТП "2" (1:1:1 -> 2:1:1)	Х Х (карта #193)
3	16.05.2023, 04:05:15	Совершён проход по разрешению внешней системы	ТП "2" (1:1:1 -> 2:1:1)	Х Х (карта #193)
2	16.05.2023, 02:38:45	Совершён проход по разрешению внешней системы	ТП "2" (1:1:1 -> 2:1:1)	Х Х (карта #193)
1	16.05.2023, 01:36:51	Совершён проход по разрешению внешней системы	ТП "2" (1:1:1 -> 2:1:1)	Х Х (карта #193)

Рисунок 24 – информация о дате и времени проходов через СКУД с данными о номере карты

СКД Страж

Персонал Карты Журнал событий **Отчёты** Администрирование доступа Настройки СКД Настройки контроллера admin (9:59)

Настройки отчётов УРВ

Отчёт УРВ за период

Дневной отчёт УРВ

Настройка экспорта данных УРВ

Экспорт событий доступа

За 16 мая 2023 г. Учитывать все события Сформировать отчёт

Фильтр по имени Начало 09:00 Окончание 18:00 Рабочий день 9:00 Применить

#	ФИО	Приход	Опоздание	Уход	Уход раньше	Отработано	Пропущено
1	X X	08:09		09:48	8:12	1:38	7:21
2	X X	09:32	0:31	09:33	8:27	0:01	8:58
3	X X	01:36		04:05	13:55	2:28	6:31

Рисунок 25 – дневной отчёт о проходах в СКУД с данными об опозданиях, пропущенном рабочем времени, времени прибытия и времени убытия

В СКУД «РУБЕЖ» находится встроенное программное обеспечение в контроллер, в котором работает веб-интерфейс, где настраивается с помощью вуб-браузера контроллер РУБЕЖ, карты, сотрудники, в том числе можно смотреть отчёты.

Таким образом, разработанная программа успешно решает проблему учета времени на перерывы сотрудников в СКУД. Результаты тестирования показали, что система успешно прошла проверку на функциональность, надежность и соответствие требованиям. Исправление обнаруженных проблем позволило улучшить работу системы.

Заключение

В результате выполнения дипломной работы была разработана система учета рабочего времени сотрудника АО «АВТОВАЗ» на основе событий системы контроля и управления доступом (СКУД). Решены две важные проблемы, которые встречаются в разработке подобных систем: проблема с интеграцией с системами учета рабочего времени и проблема с точностью и достоверностью данных в СКУД.

В процессе выполнения работы был произведен анализ требований и разработана архитектура системы учета рабочего времени с использованием событий СКУД. Исследованы и проанализированы события, регистрируемые системой контроля и управления доступом. Было выявлено, что данные, получаемые из этих событий, могут быть использованы для учета рабочего времени сотрудников. Реализованы основные функциональные модули системы, такие как модуль сбора данных, модуль обработки данных и модуль представления данных. Для обеспечения точности и достоверности данных была разработана система контроля качества данных в СКУД.

Для решения проблемы интеграции с системами учета рабочего времени была разработана программа для преобразования данных и синхронизации между системой учета рабочего времени и СКУД. Программа позволяет получать данные из системы учета рабочего времени, преобразовывать их в формат, понятный СКУД, получать данные из СКУД, объединять данные и отправлять их в СКУД.

Для решения проблемы учёта перерывов разработана и реализована программа, которая позволяет учитывать время на перерывы сотрудников в рамках СКУД. Программа основана на событиях, получаемых от системы контроля доступа, и позволяет точно определить время начала и окончания перерывов каждого сотрудника.

Выполнено тестирование основного функционала разработанных алгоритмов, результаты представлены на рисунках.

Таким образом, основываясь на требованиях к данной выпускной квалификационной работе, были разработаны эффективные решения для проблем, связанных с интеграцией с системами учета рабочего времени и точностью и достоверностью данных в СКУД, решения проблемы учёты времени перерывов. Разработанная система учета рабочего времени позволит компании АО «АВТОВАЗ» повысить эффективность управления рабочим временем сотрудников и обеспечить надежность контроля доступа в организации.

Список используемых источников

1. Бахмутов, В. Г. Системы контроля и управления доступом [Текст]: учебное пособие / В. Г. Бахмутов. – М.: Издательство Юрайт, 2018. – 197 с.
2. Руководство по проектированию систем контроля и управления доступом [Текст] / Под ред. А. В. Рахманина, А. А. Тарасова. – М.: Издательский дом «Додэка-XXI», 2017. – 512 с.
3. Первов, А. В. Проектирование систем контроля и управления доступом к объектам [Текст]: учебное пособие / А. В. Первов, В. М. Калинин, Д. В. Власов. – СПб.: БХВ-Петербург, 2019. – 240 с.
4. Кравченко, В. Ю. Системы контроля и управления доступом к информационным ресурсам [Текст]: учебное пособие / В. Ю. Кравченко, В. С. Тюленев. – М.: Интернет-Университет Информационных Технологий (ИНТУИТ), 2015. – 105 с.
5. Максимов, С. В. Системы контроля и управления доступом [Текст]: учебное пособие / С. В. Максимов. – М.: Издательство «Экзамен», 2019. – 208 с.
6. Практическое руководство по реализации системы контроля и управления доступом [Текст] / Под ред. А. С. Маргарова. – М.: Издательство «АСВ», 2016. – 320 с.
7. Раменский, В. А. Системы контроля и управления доступом. Архитектура, принципы, технологии [Текст]: учебник / В. А. Раменский, А. И. Маркин, А. А. Алексеев. – СПб.: Издательство «БХВ-Петербург», 2019. – 608 с.

8. Разработка системы учета рабочего времени на основе событий системы контроля и управления доступом [Текст] / Под ред. А. А. Иванова. – М.: Издательство «Макс Пресс», 2018. – 320 с.
9. Разработка системы учета рабочего времени на основе событий контроля и управления доступом [Текст] / Под ред. А. В. Кузнецова. – М.: Издательство «Перспект», 2017. – 352 с.
10. Разработка системы контроля и управления доступом на основе современных технологий [Текст] / Под ред. А. А. Малыгина. – М.: Издательство «Горячая Линия-Телеком», 2018. – 272 с.
11. Kothari, J., & Shaikh, F. (2017). Time and attendance management system using fingerprint biometric. *International Journal of Innovative Research in Science, Engineering and Technology*, 6(2), 1985-1989.
12. Zhang, Y., Li, Q., Li, X., & Li, Y. (2019). Design and implementation of a time and attendance management system based on face recognition. *Journal of Physics: Conference Series*, 1168(3), 032064.
13. Chawla, S., & Jain, R. (2019). Biometric authentication for time and attendance management system using Internet of Things. In *Proceedings of the 2019 3rd International Conference on Computing Methodologies and Communication (ICCMC)* (pp. 86-90). IEEE.
14. Ghosh, S., & Naskar, S. K. (2020). Time and attendance management system using RFID and IoT. In *Proceedings of the 2020 4th International Conference on Intelligent Computing and Control Systems (ICICCS)* (pp. 1742-1747). IEEE.
15. Liu, Y., Zhang, Y., Zhu, Y., & Wu, W. (2018). Design and implementation of attendance management system based on Internet of Things.

In Proceedings of the 2018 2nd International Conference on Computer Science and Application Engineering (CSAE) (pp. 20-23). IEEE.

16. Chen, Y., & Liu, Y. (2020). Design and implementation of a cloud-based attendance management system using face recognition. In Proceedings of the 2020 International Conference on Cyber-Enabled Distributed Computing and Knowledge Discovery (CyberC) (pp. 37-44). IEEE.

17. Wang, Y., & Wang, Y. (2017). Research on the design of a mobile attendance management system based on Android. In Proceedings of the 2017 International Conference on Computer and Communication Engineering (ICCCE) (pp. 13-16). IEEE.

18. Zhang, Y., Hu, Y., & Song, J. (2020). A wireless attendance management system based on RFID and ZigBee. In Proceedings of the 2020 2nd International Conference on Computer Science and Software Engineering (CSSE) (pp. 142-146). IEEE.

19. Gurjar, P. R., & Huddar, M. K. (2020). Implementation of attendance management system using face recognition with deep learning. In Proceedings of the 2020 International Conference on Intelligent Computing and Control Systems (ICICCS) (pp. 701-705). IEEE.

20. Chen, C., Chen, Y., & Ma, J. (2019). Design and implementation of an attendance management system based on GPS and Internet of Things. In Proceedings of the 2019 5th International Conference on Control Science and Systems Engineering (ICCSSE) (pp. 44-48). IEEE.

21. Swain, S. K., & Rout, R. (2017). An automated attendance management system using RFID. In Proceedings of the 2017 International Conference on Wireless Communications, Signal Processing and Networking (WiSPNET) (pp. 2406-2410). IEEE.