

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

Кафедра _____ «Прикладная математика и информатика» _____
(наименование)

09.03.03 Прикладная информатика
(код и наименование направления подготовки / специальности)

Разработка программного обеспечения
(направленность (профиль) / специализация)

ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА (БАКАЛАВРСКАЯ РАБОТА)

на тему «Разработка программного обеспечения информационной системы
контроля рабочего времени сотрудников компании»

Обучающийся

А.В.Чистяков

(Инициалы Фамилия)

(личная подпись)

Руководитель

д.т.н., доцент, С.В. Мкртычев

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

Аннотация

Тема бакалаврской работы – «Разработка программного обеспечения информационной системы контроля рабочего времени сотрудников компании».

Актуальность работы обусловлена необходимостью автоматизации учета рабочего времени персонала.

Объектом исследования является информационная система автоматизации учета рабочего времени сотрудников компании ООО «И-Сеть Телеком».

Предметом исследования выступает программное обеспечение для контроля рабочего времени сотрудников ООО «И-Сеть Телеком».

Целью данной работы является создание программного обеспечения для информационной системы учета рабочего времени, которое объединяет автоматический трекинг рабочего времени сотрудников, предоставляет интерфейс для взаимодействия с базой данных, и автоматизирует импорт данных о прохождении сотрудниками контрольных пунктов из внешней системы.

Логика и структура данной работы определяются главной целью исследования и его основными задачами. Данная работа состоит из введения, трех глав, заключения и списка использованных источников.

Практическая значимость работы заключается в возможности использования полученных результатов в деятельности ООО «И-Сеть Телеком» и других подобных организаций.

Бакалаврская работа состоит из 61 страниц текста, 18 рисунков, 5 таблиц и 20 источников.

Оглавление

Введение.....	4
Глава 1 Постановка задачи на разработку программного обеспечения для системы учёта рабочего времени сотрудников ООО «И-Сеть Телеком»	7
1.1 Обоснование необходимости контроля рабочего времени сотрудников компании	7
1.2 Разработка требований к ПО контроля рабочего времени сотрудников компании	10
1.3 Анализ известных программ контроля рабочего времени сотрудников компании	14
1.4 Анализ программно-аппаратных комплексов.....	19
1.5 Разработка требований к программному обеспечению информационной системы контроля рабочего времени сотрудников компании ООО «И-Сети Телеком»	21
1.6 Спецификация функциональных и нефункциональных требований к ПО информационной системы.....	23
Глава 2 Проектирование программного обеспечения ИС	26
2.1 Информационное обеспечение	26
2.2 Логическое моделирование	34
2.3 Техническое обеспечение	39
Глава 3 Реализация и тестирование программного обеспечения ИС	44
3.1 Реализация программного обеспечения	44
3.2 Тестирование программного обеспечения	49
Заключение	53
Список используемой литературы	55
Приложение А Фрагмент программного кода	57

Введение

От эффективности работы предприятия во многом зависит его прибыль и его успешность. Автоматизация повседневных задач является важной предпосылкой эффективной деятельности. Автоматизация учета рабочего времени (РВ) персонала - это одна из данных задач. Разумеется, замена людей машинами возможна не в каждой задаче, тем не менее, в случаях, когда это является возможным, существенно снижается количество издержек и ошибок, существенно повышается эффективность работы. На сегодняшний день наибольшая часть предприятий уже используют разные методы отслеживания рабочего времени наемного персонала. Проводится анализ полученных данных, и на основании этих данных может анализироваться эффективность деятельности работников, выносится решение о начислении им зарплаты.

Актуальность темы настоящей работы связана с тем, что в современных условиях ведения бизнеса рынок систем учета РФ представлен весьма обширной совокупностью решений. Исходя из размеров компании, реализуемых в ней бизнес-процессов, технологических ограничений и возможностей, бюджета, каждое отдельное предприятие может провести анализ, непосредственный выбор и практическое внедрение такого решения, которое в абсолютной степени будет решать задачу, связанную с учетом РВ персонала.

Задача настоящего исследования – разработка программного обеспечения информационной системы учета рабочего времени, которая бы сочетала в себе возможности автоматического трекинга времени работы сотрудников, предоставляла бы интерфейс для взаимодействия с базой данных, а также автоматизировала процесс импорта из внешней системы данных о пересечении сотрудниками пункта контроля пропуска.

Объект исследования данной работы – информационная система автоматизации учета рабочего времени сотрудников ООО «И-Сеть Телеком».

Предметом исследования является программное обеспечение

информационной системы контроля рабочего времени сотрудников ООО «И-Сеть Телеком».

Цель данной работы – разработка программного обеспечения информационной системы контроля рабочего времени сотрудников компании ООО «И-Сеть Телеком».

Для достижения поставленной цели в работе ставятся следующие задачи:

- выполнить постановку задачи на разработку программного обеспечения информационной системы контроля рабочего времени сотрудников компании ООО «И-Сеть Телеком»;
- спроектировать программное обеспечение информационной системы контроля рабочего времени сотрудников компании ООО «И-Сеть Телеком»;
- выполнить реализацию и тестирование программного обеспечения информационной системы контроля рабочего времени сотрудников компании ООО «И-Сеть Телеком».

Информационную основу исследования составили нормативно-правовые акты Российской Федерации, учебная литература, экспертные разработки российских и зарубежных ученых, опубликованные в научной литературе и периодической печати, монографические исследования, ресурсы Интернет, бухгалтерская отчетность и локальные нормативные акты ООО «И-Сеть Телеком».

Логика и структура данной работы определяются главной целью исследования и его основными задачами. Данная работа состоит из введения, трех глав, заключения и списка использованных источников.

Практическая значимость работы заключается в возможности использования полученных результатов в деятельности ООО «И-Сеть Телеком» и других подобных организаций.

Логика и структура работы определяются целью и задачами исследования. В соответствии с логикой исследования работа состоит из

введения, двух глав, заключения и списка использованных источников.

В первой главе рассмотрены теоретические аспекты детальности телекоммуникационной компании. Во-второй главе представлены проблемы и намечены пути решения проблем. Заключение содержит основные выводы по работе.

Настоящая выпускная квалификационная работа состоит из трех глав.

Первая глава является аналитической. В ней проводится анализ объекта, предоставляется его общая технико-экономическая характеристика, и выбирается комплекс задач для автоматизации. На основе этих задач определяются основные требования к информационной системе, проводится обзор аналогичных программных продуктов, существующих на рынке, и обосновываются ключевые проектные решения по информационному, техническому и программному обеспечению.

Вторая глава посвящена проектированию. В этой части раскрываются основные этапы проектирования информационной системы, начиная с выбора и обоснования модели жизненного цикла и заканчивая написанием руководства пользователя. В проектной части также описывается структура и архитектура системы, и разрабатывается информационное обеспечение, включая базу данных.

Третья глава обосновывает целесообразность разработки программного обеспечения и представляет результаты тестирования программного продукта.

Глава 1 Постановка задачи на разработку программного обеспечения для системы учёта рабочего времени сотрудников ООО «И-Сеть Телеком»

1.1 Обоснование необходимости контроля рабочего времени сотрудников компании

Компания «И-Сеть Телеком» является одной из крупнейших телекоммуникационных компаний. Она выступает поставщиком услуг в области широкополосного доступа в интернет (ШПД), кабельного и HD-телевидения, фиксированной телефонной связи. Головной офис компании находится в г. Екатеринбург.

Компания осуществляет деятельность на базе собственных сетей, построенных по единым стандартам с использованием технологии FTTH (оптика до здания). В настоящий момент «И-Сеть Телеком» является крупным интернет-провайдером Свердловской области, входит в список крупнейших операторов кабельного телевидения. Также компания занимает лидирующие позиции по темпам подключения абонентов. Услуги компании предоставляются под торговой маркой «К-Телеком» почти в 60 городах и трех областях России. Компания «К-Телеком» образована в 2008.

Компания «И-Сеть Телеком» ориентирована на потребителей, сохранение длительных взаимоотношений с партнерами и сотрудниками, соблюдение высоких стандартов качества продукта и сервиса. Основные ценности компании – надежность, удобство и комфорт клиентов.

«В соответствии со стратегией развития компании, предполагается, что к 2025 г. бизнес-показатели должны удвоиться. «И-Сеть Телеком» и его региональные операторы добиваются выполнения этой задачи не только на основе расширения сети, ее экстенсивного роста, но и путем поглощений более мелких региональных операторов» [2].

Генеральной целью «И-Сеть Телеком» является создание в российских

городах телекоммуникационных сетей, которые будут интегрированы с системами жизнедеятельности муниципального образования и дадут возможность повысить уровень жизни местного населения, уровень информационного обеспечения городских жителей на принципиально новый уровень за счет оказания полного комплекса инновационных услуг.

К факторам успеха в достижении целей развития «И-Сеть Телеком» можно отнести:

- оптимальное техническое решение;
- технологию быстрого захвата и удержания лидерских позиций на телекоммуникационном рынке;
- ориентацию на мировые стандарты качества обслуживания: сервис/клиентоориентированность;
- стандартизацию бизнес-процессов;
- наличие высокоэффективной команды специалистов и менеджеров.

В целом, компания «И-Сеть Телеком» ориентирована на использование новейших технологий во всех сферах деятельности как для внешних потребителей услуг, так и для сотрудников.

Организация ООО «И-Сеть Телеком» зарегистрирована 08 февраля 2016 года по адресу 624006, Свердловская область, Сысертский район, поселок Большой Исток, улица Metallистов, дом 1. Компании был присвоен ОГРН 1169658012864 и выдан ИНН 6685107088. Основным видом деятельности является деятельность по предоставлению услуг телефонной связи. Оказание услуг телефонной связи - основной вид деятельности. К дополнительным видам экономической деятельности относятся:

- торговля оптовая неспециализированная;
- торговля розничная аудио- и видеотехникой в специализированных магазинах;
- торговля розничная пиломатериалами в специализированных

магазинах;

- торговля розничная вне магазинов, палаток, рынков;
- деятельность в области документальной электросвязи [13].

Учитывая расширение сфер и регионов деятельности организации, все большую значимость приобретают системы контроля рабочего времени сотрудников.

«Системы учета рабочего времени представляют собой мощный набор инструментов, обладающих значительным потенциалом для обеспечения своевременного выполнения сотрудниками своих обязанностей.

Основная задача этих систем – собирать максимально возможную информацию о деятельности сотрудников и, на основе анализа этой информации, оценивать их эффективность, что существенно снижает временные затраты.

Для многих предпринимателей проблемой может быть не только неэффективная работа сотрудников, но и недобросовестные сотрудники, которые могут пытаться заработать нечестным способом, например, продавая конфиденциальную информацию.

Если не контролировать или игнорировать это, злоумышленники могут нанести значительный ущерб развитию предприятия. Поэтому руководитель должен понимать, как работают его сотрудники, и в этом ему помогает система учета рабочего времени, которая отслеживает и оценивает эффективность их работы» [9].

В итоге, основная цель контроля рабочего времени заключается в сокращении временных затрат и защите конфиденциальной информации.

Система учета рабочего времени фиксирует действия сотрудников, такие как приход на работу, обед, уход с работы, нахождение на рабочем месте или прогулы.

С её помощью руководитель может следить за ходом выполнения работы и управлять процессом.

1.2 Разработка требований к ПО контроля рабочего времени сотрудников компании

Для выполнения указанных функций на предприятии рекомендуется использование программного обеспечения системы контроля и управления доступом (СКУД).

Наиболее подходящие СКУД-системы предоставляют следующий функционал:

- отслеживание прихода/ухода сотрудников через двери или турникеты, системы оборудованы считывателями карт, биометрическими сканерами или другими устройствами идентификации для регистрации входов и выходов сотрудников;
- разработка функции ведения журнала событий входа и выхода сотрудников с детализацией истории их перемещений. Данные сохраняются по дням;
- выдача и лишение уровня доступа по желанию руководителя, администратор системы может легко изменять уровни доступа сотрудников, добавляя или убирая их привилегии;
- запись ключей доступа в контроллеры, ключи доступа сотрудников записываются в контроллеры системы, что позволяет быстро и удобно управлять доступом;
- некоторые системы контроля и управления доступом (СКУД) могут быть интегрированы с программным обеспечением 1С. Это позволяет настраивать права доступа сотрудников в соответствии с бизнес-правилами, которые реализованы в 1С;
- учет рабочего времени по данным проходов сотрудников;
- система учитывает рабочее время сотрудников на основе информации о том, когда они приходят и уходят;

- программное обеспечение анализирует эти данные и рассчитывает продолжительность рабочего времени каждого сотрудника. Затем система автоматически заполняет обязательные таблицы учёта рабочего времени, используя информацию о перемещениях сотрудников.

Преимущества СКУД-систем:

- повышение безопасности: Контроль доступа позволяет предотвращать несанкционированные проникновения на территорию предприятия;
- повышение дисциплины: регистрация времени прихода и ухода сотрудников способствует соблюдению рабочего распорядка;
- автоматизация процессов: автоматическое ведение журналов и таблиц экономит время и снижает риск ошибок;
- потенциальные затраты на адаптацию;
- аппаратные расходы: закупка и установка турникетов, дверных замков, считывателей карт или биометрических сканеров;
- программное обеспечение: приобретение лицензий на СКУД и возможная доработка ПО для интеграции с 1С;
- техническая поддержка и обучение: обучение сотрудников работе с новой системой и услуги техподдержки [3].

Для минимизации затрат и успешного внедрения системы рекомендуется провести предварительный аудит текущих процессов и определить конкретные требования предприятия. Кроме того, важно выбрать СКУД, которая будет максимально соответствовать потребностям предприятия и легко интегрироваться с уже используемыми информационными системами [10].

Программа в течение рабочего дня автоматически и в режиме реального времени фиксирует, когда сотрудники проходят через турникет, отмечая их приход и уход.

Для предприятия программа выполняет следующие задачи:

- отслеживает, когда сотрудники входят на территорию и покидают её через двери или турникеты;
- сохраняет историю приходов и уходов сотрудников по дням;
- позволяет предпринимателю предоставлять или ограничивать доступ сотрудников;
- записывает ключи доступа в контроллеры;
- управление доступом сотрудников с помощью сценариев, написанных на языке 1С;
- учет рабочего времени на основе данных о проходах сотрудников;
- автоматическое заполнение табеля учета рабочего времени данными о приходах и уходах сотрудников.

Из перечисленных функций видно, что программа успешно справляется с основной задачей всех подобных систем – учетом рабочего времени сотрудников, обеспечивая информационную поддержку предприятия [4].

Однако для использования данной программы могут потребоваться незначительные изменения, обусловленные особенностями учета и контроля рабочего времени конкретного предприятия.

Это может привести как к незначительным, так и к значительным денежным затратам на адаптацию системы.

На рисунке 1 представлена контекстная диаграмма бизнес-процесса «Учет рабочего времени сотрудника» в организации ООО «И-Сеть Телеком».

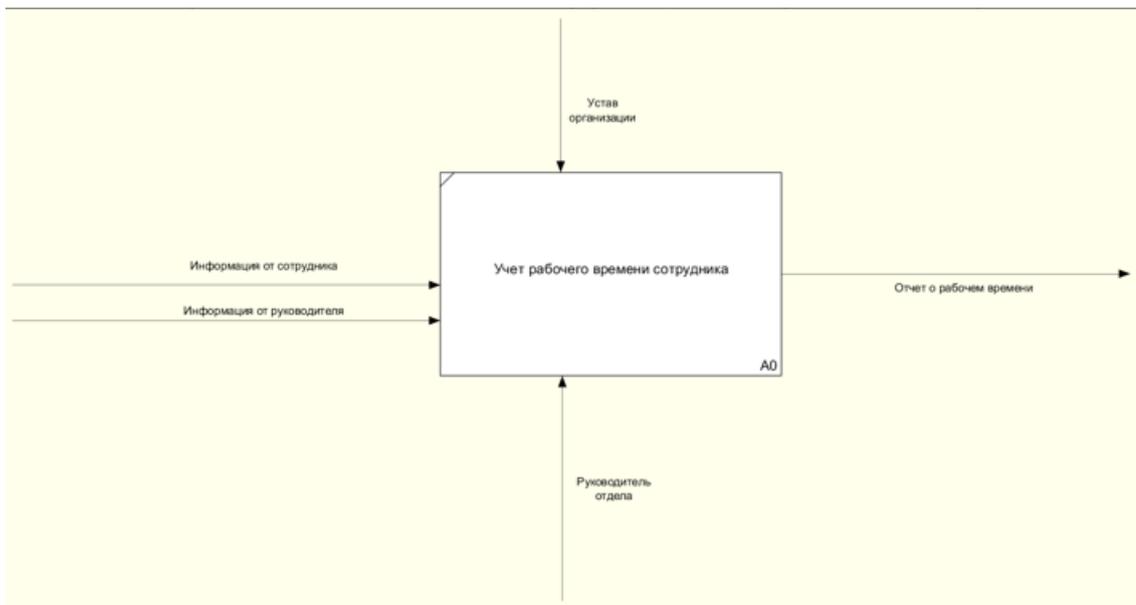


Рисунок 1 – Контекстная диаграмма бизнес-процесса «Учет рабочего времени сотрудника»

Декомпозиция контекстной диаграммы приведена на рисунке 2. Здесь выделены следующие функциональные блоки: «сбор информации», «занесение информации в таблицу», «формирование отчета о рабочем времени».

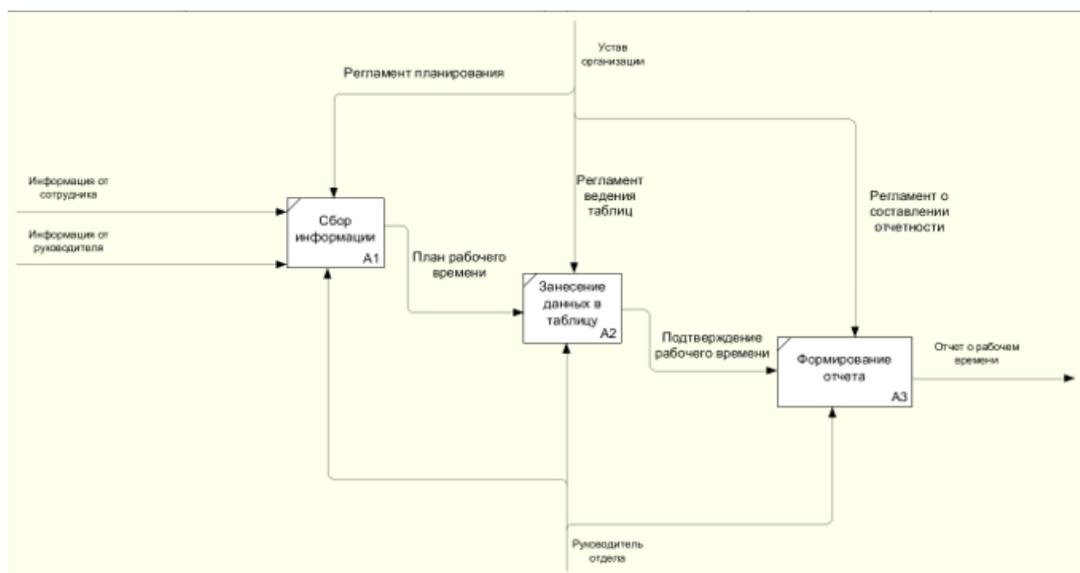


Рисунок 2 – Декомпозиция контекстной диаграммы

В процессе декомпозиции, функциональный блок, который в контекстной диаграмме отображает систему как единое целое, подвергается детализации на другой диаграмме. Получившаяся диаграмма второго уровня содержит функциональные блоки, отображающие главные подфункции функционального блока контекстной диаграммы и называется дочерней (Child diagram) по отношению к нему (каждый из функциональных блоков, принадлежащих дочерней диаграмме соответственно называется дочерним блоком – Child Box).

1.3 Анализ известных программ контроля рабочего времени сотрудников компании

Первая система, которую мы рассмотрим, называется DeskTime. Это онлайн-сервис, который выделяется своей простотой использования: достаточно установить приложение на устройство, что делает его удобным и доступным для большинства начинающих предпринимателей. Этот сервис позволяет руководителям автоматически отслеживать работу сотрудников в режиме онлайн.

Основная цель DeskTime – повышение производительности и помощь командам и индивидуальным предпринимателям в достижении желаемых результатов быстрее и эффективнее через автоматическое отслеживание. Стоит отметить, что данный сервис не увеличивает количество времени, а помогает более рационально его планировать и использовать, что является отличным начальным шагом для дальнейшего развития.

На рисунке 3 представлено главное меню ПО DeskTime.

позволяет создавать отчеты, показывающие, какие сотрудники вносят значительный вклад в развитие компании, а чьи методы требуют улучшения;

- учет рабочего времени для удаленных сотрудников. Правильный контроль работы удаленных сотрудников может повысить эффективность и снизить затраты для предприятия;
- функция свободного времени. Руководитель может выделять сотрудникам свободное время для отдыха, в течение которого слежение за ними прекращается;
- изменение рабочего времени. В периоды командировок, отпусков или на выходных сотрудник может временно приостановить учет рабочего времени» [8].

Рассмотрим систему учета рабочего времени CrocoTime.

Следующая система, которую мы рассмотрим, называется CrocoTime.

Как и предыдущая, эта система также является автоматизированной. Её основная функция заключается в автоматическом мониторинге сотрудников, работающих на своих рабочих устройствах.

На рисунке 4 изображено главное меню, с отчетами сотрудников отделов предприятия.



Рисунок 4 – Главное меню программного продукта CrocoTime, содержащее отчеты сотрудников отделов предприятия

«Основные задачи, решаемые с помощью CrocoTime:

- отслеживание активного времени. Программа фиксирует действия сотрудника в течение рабочего дня, анализирует результаты работы и формирует статистику по загруженным участкам времени;
- фотоотчет. Эта функция позволяет руководителю контролировать деятельность сотрудников, фиксируя, сколько времени они потратили на выполнение заданий и какие программы использовали. Это особенно полезно для мониторинга сотрудников, работающих дистанционно;
- анализ процессов. Руководитель может создать отчет по использованным программам и сайтам для анализа бизнес-процессов предприятия, что облегчает контроль за эффективностью работы;
- функция слежения за временем. Эта функция необходима для учета рабочего времени, расчета времени, затраченного на выполнение бизнес-проектов и задач, а также для оценки выполненной работы;
- таблицы учета. Программа позволяет автоматически заполнять таблицы учета рабочего времени сотрудников, учитывая график работы, выходные, прогулы и больничные, что сокращает необходимость ручного ввода данных» [8].

Система учета рабочего времени «СКУД и УРВ для 1С: Предприятие 8»

В качестве рассматриваемого программного продукта была выбрана система «СКУД и УРВ для 1С: Предприятие 8». Эта программа, как и предыдущие продукты, предназначена для учета рабочего времени сотрудников на предприятии и представляет собой внешнюю обработку на базе технологической платформы «1С: Предприятие» версий 8.2 и 8.3.

Программа отслеживает время прихода и ухода сотрудников на работу, считывая эти данные при прохождении через турникет. Важно, чтобы программное обеспечение и техническое оборудование были совместимы друг с другом. Для выбора комплектующих необходимо ознакомиться с документацией производителя.

Для лучшего понимания принципа работы системы учета рабочего времени, на рисунке 5 представлена общая схема работы.

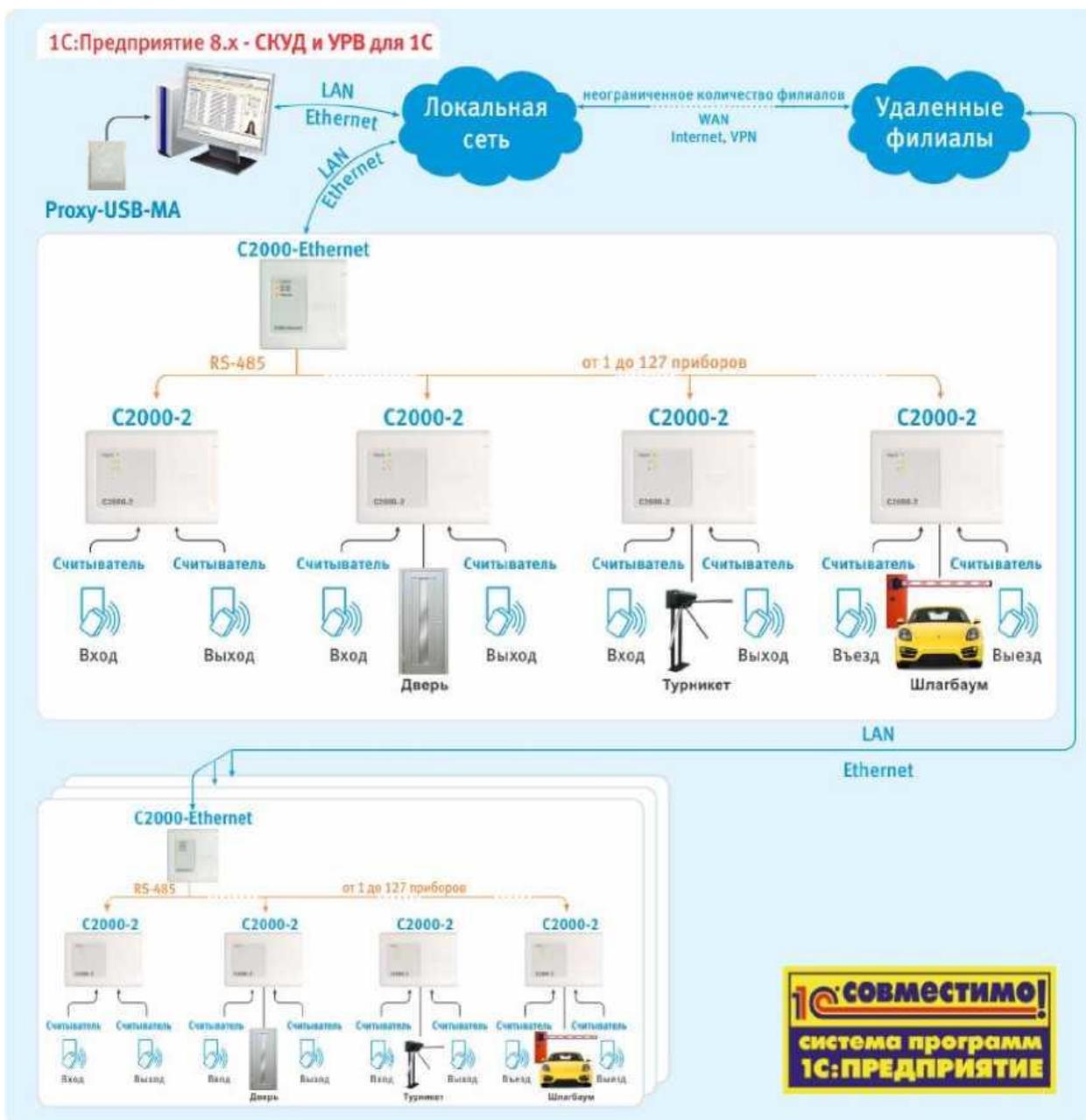


Рисунок 5 – Общая схема работы системы учета рабочего времени

В отличие от ранее рассмотренных систем учета рабочего времени данная программа имеет свои характерные плюсы и минусы

1.4 Анализ программно-аппаратных комплексов

В ООО «И-Сеть Телеком» одним из основных недостатков учета рабочего времени является его полное ведение вручную. Работники и руководители подразделений часто забывают или искажают информацию о времени прихода и ухода с рабочего места. С ростом числа сотрудников становится невозможно контролировать рабочее время вручную. В качестве решения была разработана рекомендация по созданию автоматизированного сервиса для учета рабочего времени с пользовательским интерфейсом. Это позволит ООО «И-Сеть Телеком» без существенных затрат автоматизировать учет рабочего времени и получить точную информацию об эффективности работы сотрудников.

Чтобы решить эту задачу, нужно проанализировать готовые ИТ-проекты, уже существующие на текущий момент времени, которые успешно работают на предприятиях, функционирующих в сфере торговли.

В настоящее время отмечается значительное разнообразие систем учета, программных продуктов, которые позволяют автоматизировать РВ персонала.

Рынок систем учета РВ сотрудников предлагает обширный спектр разных решений, и возможна их адаптация под потребности каждого предприятия. Рассмотрим лишь некоторые из них:

PERCo - ведущий российский производитель систем и оборудования безопасности. Оборудование PERCo активно внедряется по всей России и экспортируется в 73 страны мира. СКУД PERCo предназначена для защиты от проникновения посторонних, разграничения доступа сотрудников внутри предприятия, автоматизации учета рабочего времени. Perco построена на основе объединённых в сеть посредством Ethernet контроллеров и компьютеров. Контроллеры подключаются к считывателям, которые обрабатывают информацию с таких идентификаторов, как бесконтактные (Proximity) карты и брелоки стандарта EM-Marin, HID, Mifare. PERCo предлагает клиентам собственные УПУ: турникеты различных видов,

электромеханические замки. Использование всех элементов одного производителя позволяет повысить совместимость устройств и надежность всей системы.

Орион. Продукция «Болид», интегрированная система охраны «Орион» – это комплекс программных и аппаратных средств, необходимых для создания систем автоматического пожаротушения, видеонаблюдения, контроля доступа, охранно-пожарной сигнализации, и, вместе с тем, для построения систем контроля, а также диспетчеризации объектов.

Это система идёт далее, нежели обычные СКУД. Она представляет собой комплексное решение в сфере безопасности для большинства необходимых автоматизаций любого предприятия [7].

Представим сравнительную характеристику программно-аппаратных комплексов в виде аналитической таблицы 1.

Таблица 1 – Сравнительная характеристика программно-аппаратных комплексов

Параметры сравнения	PERCo	ИСО «Орион Про»	Самостоятельная разработка
Назначение	Система контроля доступа для малых и средних предприятий	Система контроля доступа для крупных и средних предприятий	Система контроля доступа для малых, средних и крупных предприятий
Программное обеспечение	Требуется установка только сервера системы.	АРМ "Орион Про"	ПО, разработанное сотрудниками компании
Удаленный доступ	Из любого места при подключении к интернету	Из любого места при подключении к интернету	Из любого места при подключении к интернету
Контроль доступа транспорта	Нет	Нет	Да
Видеонаблюдение	Нет	Да	Да

Продолжение таблицы 1

Параметры сравнения	PERCo	ИСО «Орион Про»	Самостоятельная разработка
Охранно-пожарная сигнализация	Нет	Да	Да
Центральный пост охраны	Нет	Нет	Да

Любая из приведенных программ имеет свои преимущества. Эти программы включают в себя всю совокупность функций по учету РВ. Тем не менее, имеются у данных программных средств некоторые недостатки, а именно: широкий функционал данных программных средств останется неиспользованным. Более того, их приобретение, внедрение, поддержка и сопровождение связаны с высокими финансовыми затратами, что в текущей экономической ситуации компании совершенно неприемлемо. Таким образом, приобретение проанализированного программного обеспечения является нецелесообразным.

Таким образом, было решено создать собственную систему учета рабочего времени, более функциональную и простую.

1.5 Разработка требований к программному обеспечению информационной системы контроля рабочего времени сотрудников компании ООО «И-Сети Телеком»

Цель автоматизации рассматриваемого процесса контроля рабочего времени включает:

- автоматизацию составления отчетов о проделанной работе;
- обеспечение работоспособности сервера системы;
- исключение ошибок при подсчете статистики;
- обеспечение прозрачного получения результатов работы;
- оперативный обмен информацией между всеми участниками аналитической работы;

- возможность для руководства контролировать отработанное сотрудниками время;
- способность передавать отчеты в государственные службы статистики.

Требования к функциональности автоматизированной информационной системы (АИС) включают:

- автоматический сбор необходимой информации о действиях сотрудников путем анализа системных файлов рабочего места;
- полное отказ от бумажных версий документов с хранением всех учетных документов в электронном виде;
- прозрачная регистрация действий сотрудников, анализ и составление отчетов;
- предоставление экранных форм для просмотра результатов работы, печати отчетов и отправки их по электронной почте;
- включение как минимум электронного табеля учета отработанного времени в состав учетных документов, с возможностью добавления других по мере необходимости.

Требования к архитектуре и реализации АИС включают:

- использование баз данных в СУБД MS SQL;
- возможность резервного копирования и восстановления данных;
- ежедневное обновление информации в учетных документах;
- возможность масштабирования системы и ее простая установка в других организациях с схожими профилями после успешной реализации.

«Методология проектирования информационных систем описывает процесс создания и сопровождения систем в виде жизненного цикла (ЖЦ) ИС, представляя его как некоторую последовательность стадий и выполняемых на них процессов. Для каждого этапа определяются состав и последовательность выполняемых работ, получаемые результаты, методы и средства,

необходимые для выполнения работ, роли и ответственность участников и т. д. Такое формальное описание ЖЦ ИС позволяет спланировать и организовать процесс коллективной разработки и обеспечить управление этим процессом» [6].

1.6 Спецификация функциональных и нефункциональных требований к ПО информационной системы

Основные требования к системе:

- обеспечение хранения в БД сведений о рабочем времени;
- формирование электронных форм и документов для отчетности;
- формирование и печать отчетов о рабочем времени сотрудников;
- обеспечение ведения работы в системе с различными правами доступа.

«Интерфейс системы должен быть простым, понятным и удобным для пользователя, обеспечивать быстрое отображение экранных форм. Навигационные элементы должны быть выполнены в удобной для пользователя форме» [3].

«Функциональные требования описывают, что необходимо реализовать в продукте или системе. Они содержат ту ценность системы, ради которой она создаётся – логику, взаимодействие её компонентов и пользователей с ней.

Нефункциональные требования (НФТ) описывают, как должен работать программный продукт и какими свойствами или характеристиками обладать, чтобы доставить ту ценность, которую несёт система, с учетом условий ее существования» [8].

«Для модульной структуры пакет программ состоит из набора функциональных модулей и утилит. Пользователь имеет возможность выбрать для каждого объекта индивидуальный состав модулей и их количество, гибко настраивать каждое рабочее место.

В системе с клиент-серверной архитектуры можно организовать

полноценное автоматизированное рабочее место на одном компьютере либо создать распределенную сеть из более 200 рабочих мест, связанных по Ethernet или VPN-каналу. Допускается наращивание системы за счет приобретения дополнительных модулей в процессе эксплуатации» [1].

В системах подобной архитектуры пользователи могут получить доступ к приложению через уровень представления, который взаимодействует либо напрямую с компонентами бизнес-уровня; или через фасад приложения на бизнес-уровне, если методы связи требуют объединения функциональных возможностей. Между тем, внешние клиенты и другие системы могут получить доступ к приложению и использовать его функциональные возможности, взаимодействуя с бизнес-уровнем через сервисные интерфейсы. Это позволяет приложению лучше поддерживать несколько типов клиентов и способствует повторному использованию и более высокому уровню функциональности в приложениях [5].

В некоторых случаях уровень представления может взаимодействовать с бизнес-уровнем через уровень сервисов. Однако это не является абсолютным требованием. Если при физическом развертывании приложения уровень представления и бизнес-уровень находятся на одном уровне, они могут взаимодействовать напрямую [18].

Решаемые задачи представлены ниже.

Организация рабочего места оператора систем безопасности с выводом информации на монитор, отображением графического плана охраняемых помещений, возможностью интерактивной постановки множества объектов на охрану и снятия с охраны в «два клика», а также управления точками доступа

Формирование рабочего места постового на проходной с идентификацией посетителей по фотографии [1].

Обеспечение на программном уровне любых взаимосвязей между подсистемами безопасности, не доступных на релейном уровне.

Выводы по главе 1

В первой главе проанализирована деятельность компании ООО «И-Сети Телеком». Указаны основные результаты, направления и цели компании. Описана сфера деятельности и стратегия развития компании. Выявлены цели и задачи компании по применению результатов контроля рабочего времени сотрудников.

Определены основные функции, выполняемые разрабатываемым программным обеспечением.

Описаны существующие бизнес-процессы учета работы сотрудников компании и выбран комплекс процессов, подлежащий автоматизации.

Проведен анализ существующего ПО по учету рабочего времени.

Проведена постановка задачи на разработку и определены требования к программному обеспечению.

Рассмотрен план построения информационной системы со стороны построения архитектуры программного обеспечения

Глава 2 Проектирование программного обеспечения ИС

2.1 Информационное обеспечение

Инфологическая модель БД – это описание сущностей (объектов), с комплексом атрибутов и связей между ними, выявляющихся в ходе изучения не только входных данных, но и выходных. Основное предназначение инфологической модели базы данных состоит в следующем: структурное создание предметной области, с ориентацией на информационное внимание пользователей создаваемой ИС. Необходимо, чтобы она была не только стабильной, но и неизменной. Инфологическая модель – это представление аспекта пользователя создаваемой системы на раннее описанную предметную область. Тем не менее, в ходе проектирования инфологической модели базы данных, необходимо, чтобы имелась возможность для увеличения этой модели, а также для вставки вспомогательных информационных данных.

Модель "сущность-связь" является в инфологическом моделировании наиболее распространенной моделью. Сущности и связи являются основными компонентами этой модели. Сущность – это содержание объекта, о котором набирается требуемая информация. Чёткий объект представляется экземпляром сущности. Определяется она атрибутами, описанными соответствующими характеристиками. Необходимо, чтобы связи показывать имеющиеся между сущностями определенные отношения. При создании инфологической модели, как правило, применяются графические схемы.

Самым распространенным представителем класса семантических моделей предметной области (инфологических, концептуальных) является ER-модель или модель «сущность-связь». ER-модель представляет собой способ описания структуры базы данных в графической форме с использованием ER-диаграммы, разработанной П. Ченом, либо другими графическими нотациями, такими как Information Engineering или Crow's Foot.

Основные преимущества ER-моделей включают:

- наглядность представления структуры данных;
- возможность проектировать базы данных с большим количеством объектов и атрибутов;
- поддержка в различных системах автоматизированного проектирования баз данных, таких как ERWin.

Главные элементы ER-моделей включают:

- объекты (сущности) предметной области;
- атрибуты этих объектов;
- связи между объектами.

Сущность предметной области – это объект, который имеет определенные атрибуты.

Связь между сущностями характеризуется:

- типом связи (например, один к одному, один ко многим, многие ко многим);
- классом принадлежности (какая сторона связи является обязательной или необязательной для существования записи в другой сущности).

Стоит отметить, что класс может являться, как обязательным, так и необязательным. В случае, если в связи участвует каждый экземпляр сущности, то тогда класс принадлежности является обязательным, в противоположной ситуации – необязательным.

В работе используется база данных MySQL. Речь идет о самой распространенной на всей планете системе управления реляционными БД с лицензией GPL (General Public License). Эта реляционная система управления базами данных широко используется во всем мире благодаря своей надежности, высокой производительности и простоте в использовании. Она стала предпочтительным выбором для веб-приложений, используемых на крупных веб-ресурсах, таких как Facebook, Twitter, YouTube, Yahoo! и многих других.

MySQL является свободным программным обеспечением с открытым исходным кодом в соответствии с условиями General Public License, а также доступно под различными частными лицензиями. MySQL принадлежал и спонсировался шведской компанией MySQL AB, которая была куплена Sun Microsystems (сейчас Oracle Corporation).

Логическая модель БД изображена на рисунке 6.

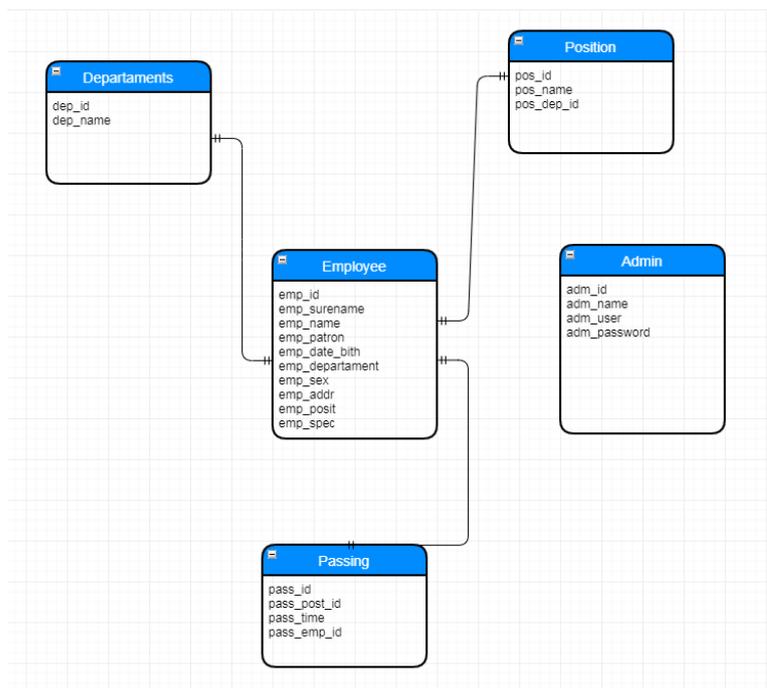


Рисунок 6 – Логическая модель БД

Модель представлена следующими таблицами:

- Passing (контрольно-пропускной пункт);
- Departaments (отделы);
- Position (должность);
- Employee (сотрудники).

Даталогическая модель базы данных описывает логическую структуру данных без привязки к конкретному способу их хранения или содержанию. Это представление устанавливает связи между данными на уровне информационных единиц, которые поддерживает конкретная система

управления базами данных (СУБД). Создание такой модели называется даталогическим или логическим проектированием.

В таблице 2 показано описание структуры таблиц БД.

Таблица 2 – Описание структуры таблиц БД

Сущность	Идентификатор таблицы	Атрибут	Идентификатор поля	Тип поля
Сотрудники	EMPLOYEE	ID сотрудника	EMP_ID	integer
		Имя сотрудника	EMP_NAME	varchar
		Фамилия сотрудника	EMP_SURENAME	varchar
		Отчество сотрудника	EMP_PATRON	varchar
		Дата рождения	EMP_DATE_BIRTH	date
		Отдел	EMP_DEPARTAMENT	integer
		Должность	EMP_POSIT	integer
		Пол	EMP_SEX	integer
		Адрес	EMP_ADDR	varchar
		Специальность	EMP_SPEC	varchar
Отделы	DEPARTAMENTS	ID отдела	DEP_ID	integer
		Название отдела	DEP_NAME	varchar
Должности	POSITION	ID должности	POS_ID	integer
		Название должности	POS_NAME	varchar
		ID отдел	POS_DEPARTAMENT	integer
Контрольно-пропускной пункт	PASSING	ID записи	PASS_ID	integer
		ID сотрудника	PASS_EMP_ID	integer
		ID Контрольно-пропускного пункта	PASS_POST_ID	integer
		Время прохождения	PASS_TIME	date
Администратор	ADMIN	ID администратора	ADM_ID	int
		Имя администратора	ADM_NAME	varchar
		Логин администратора	ADM_USER	varchar
		Пароль администратора	ADM_PASSWORD	varchar

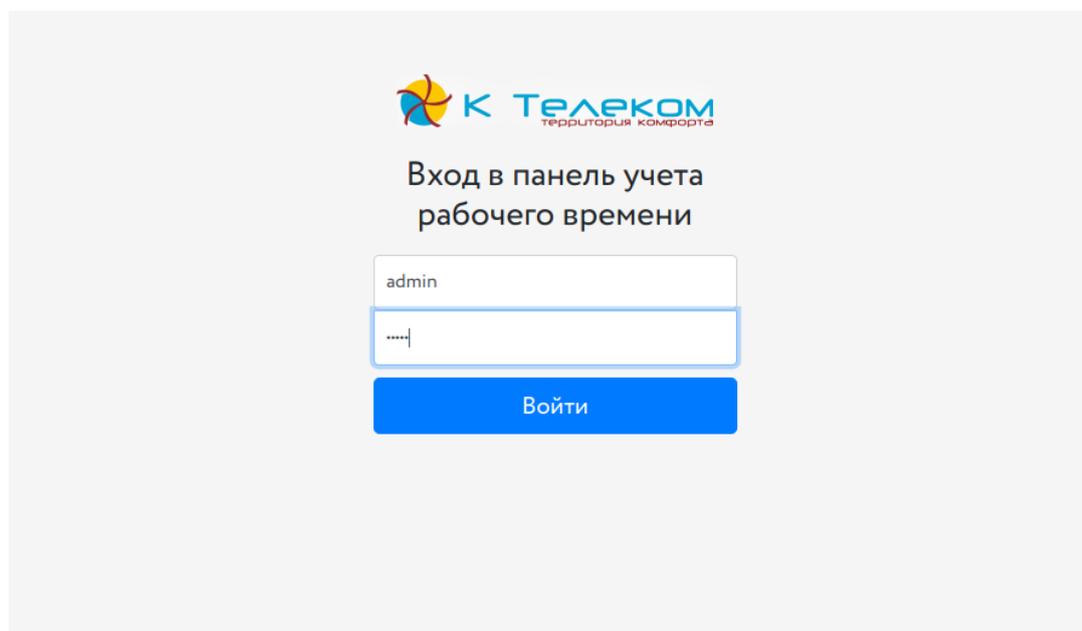
Входная информация (ВИ) – это любая информация, которая нужна для решения задачи, она располагается на разных носителях: в памяти персонального компьютера, в машинных носителях, в первичных документах. Для этого создается перечень ВИ, состав реквизитов каждого отдельного вида ВИ, расположение реквизитов ВИ, описание реквизитов (полей) входных документов.

Информация - это сведения, сообщения о каком-либо событии, деятельности и т.д. Под информацией следует понимать все те сведения, которые необходимо фиксировать, передавать, хранить и обрабатывать.

ВИ – это любая информация, которая требуется для решения задачи, она располагается на разных носителях: в памяти персонального компьютера, в машинных носителях, в первичных документах.

Информация, вводимая администратором (рисунки 7,8):

- логин и пароль;
- дата начала отчета;
- дата окончания отчета;
- фамилия сотрудника.



The image shows a login interface for 'К ТЕЛЕКОМ' (K TELECOM) with the tagline 'территория комфорта'. The title of the form is 'Вход в панель учета рабочего времени' (Login to the working time accounting panel). There are two input fields: the first contains the username 'admin', and the second is for the password, shown as dots. Below the fields is a blue button labeled 'Войти' (Login).

Рисунок 7 – Авторизация

Рабочее время сотрудников

Дата начала Дата окончания [Показать](#)

Фамилия сотрудника

[День](#) [Месяц](#) [Год](#)

Рисунок 8 – Ввод параметров отчета

Выходная (результативная) информация является целью обработки планово-учётных данных и представляет собой сведения, полученные в процессе решения задач, и используется работниками для планирования, учёта, контроля, анализа, принятия решений.

Выходными данными являются отчеты. При выборе промежутка времени равного суткам, будет сформировано два отчета. Первый отчет содержит в себе поля:

- дата;
- отдел;
- сотрудник;
- должность;
- время прихода;
- время ухода;
- время, проведенное на работе.

Второй отчет содержит поля:

- дата;
- отдел;
- сотрудник;
- должность;
- переработка;

- недоработка.

При выборе промежутка времени равного двум суткам и более, так же будет сформировано два отчета, но данные отчета будут суммированы заданному периоду. Таким образом можно составить месячный, квартальный, годовой отчеты. Первый отчет содержит поля:

- отдел;
- сотрудник;
- должность;
- время, проведенное на работе.

Второй отчет содержит поля те же самые поля, как и при выборе промежутка равном суткам за исключением поля дата (рисунки 9,10).

Дата	Отдел	Сотрудник	Должность	Пришел	Ушел	Время на работе
2022-05-29	IT	Быкова Мариша	Системный администратор	08:22:55	20:22:35	11:59:40
2019-05-29	Отдел администрирования	Филатова Анастасия	Стажер	08:20:50	19:22:35	11:01:45
2022-05-29	Финансово-экономический отдел	Виноградова Ульяна	Экономист	10:22:55	18:02:35	07:39:40

Рисунок 9 – Отчет «Рабочее время сотрудников»

Дата начала Дата окончания [Показать](#)
[День](#) [Месяц](#) [Год](#)
 Фамилия
 сотрудника

[Экспорт в Excel](#)

Рабочее время Переработка

Дата	Отдел	Сотрудник	Должность	Переработка	Недоработка
2022-05-29	IT	Быкова Мариша	Системный администратор	03:59:40	
2022-05-29	Отдел администрирования	Филатова Анастасия	Стажер	03:01:45	
2022-05-29	Финансово-экономический отдел	Виноградова Ульяна	Экономист		00:20:20

Рисунок 10 – Отчет «Отклонение от нормы рабочего времени»

Также любой отчет можно экспортировать в файл формата .xls для работы с отчетом в программе Microsoft Excel (рисунок 11).

	A	B	C	D	E	F	G
1	Дата	Отдел	Сотрудник	Должность	Пришел	Ушел	Время на работе
2	29.05.2022	IT	Быкова Мариша	Системный администратор	8:22:55	20:22:35	11:59:40
3	29.05.2022	Отдел администрирования	Филатова Анастасия	Стажер	8:20:50	19:22:35	11:01:45
4	29.05.2022	Финансово-экономический отдел	Виноградова Ульяна	Экономист	10:22:55	18:02:35	7:39:40

Рисунок 11 – Экспорт отчета в Microsoft Excel

Таким образом предоставляемые входные и выходные данные в виде экранных и печатных форм ввода и вывода информации визуально покрывают составленные нами требования к системе [14].

2.2 Логическое моделирование

UML-диаграмма представляет собой специализированный графический язык для моделирования объектов в разработке программного обеспечения. Этот язык использует различные графические символы для создания абстрактной модели системы и включает в себя широкий набор стандартных обозначений.

Цель UML состоит в том, чтобы поддерживать определение, визуализацию, документирование и проектирование различных программных систем. Важно отметить, что UML не является языком программирования, но предусматривает возможность автоматической генерации кода на основе диаграмм.

Помимо программного обеспечения, UML активно применяется для моделирования бизнес-процессов, системного проектирования и отображения организационных структур. Разработчики могут использовать UML для согласованного представления общих концепций, таких как компоненты, классы, наследование, поведение и агрегация, что способствует более высокому уровню внимания к архитектуре и проектированию систем.[16]

Прецедентная модель процессов отражает основные процессы и их взаимодействие с окружением.

Начинается с построения внешней диаграммы (Use Case Diagram), показывающей, как процесс виден извне.

Актор (действующее лицо) – субъект окружения процесса.

Прецедент (вариант использования) – относительно законченная последовательность действий в рамках некоторого бизнес-процесса, приносящая ощутимый результат конкретному автору (рисунок 12).



Рисунок 12 – Диаграмма прецедентов

Диаграмма классов описывает разновидности классов в системе и их статические связи.

На таких диаграммах изображаются атрибуты классов, их методы и ограничения на связи.

Интерпретация диаграммы зависит от уровня абстракции: классы могут представлять сущности предметной области или элементы программной системы.

Основные элементы включают классы с их атрибутами и методами. Атрибуты описывают свойства объектов класса, а операции представляют собой функции или преобразования, которые могут принимать параметры и возвращать значения (рисунок 13) [12].

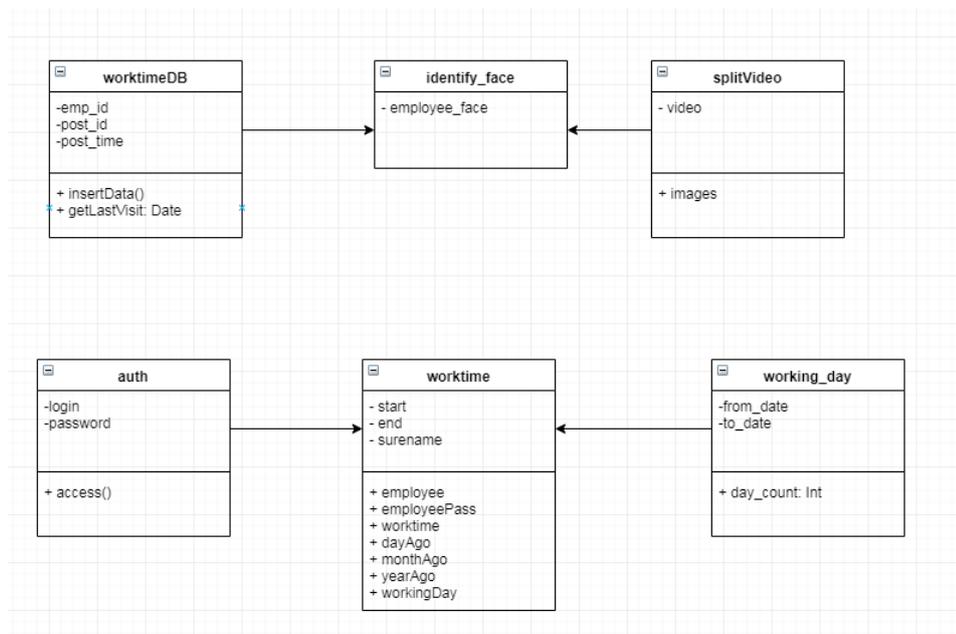


Рисунок 13 – Диаграмма классов

Класс `identify_face` занимается распознаванием человека на видео. Для того, чтобы распознавание было возможно, необходимо загрузить фото сотрудников. Сотрудники предоставляют видео с записью своего лица с разных ракурсов длительностью 15-20 секунд. Класс `splitVideo` разделяет видео каждого сотрудника покадрово на множество фотографий. После того как распознавание начато и приложение распознало сотрудника, необходимо проверить, когда последний раз сотрудник проходил через контрольно-пропускной пункт. Если программа зарегистрировала проход через один и тот же КПП несколько раз за минуту, вероятнее всего, сотрудник еще не успел пройти. Для того, чтобы не определять одно и тоже пересечение КПП несколько раз используется класс `worktimeDB`, который возвращает последнее время прохождения КПП сотрудника и заносит в базу данных время прохождения после успешной идентификации.

Класс `auth` дает доступ к веб-приложению с отчетами о рабочем времени сотрудников после авторизации по логину и паролю.

После доступа к приложению администратор с помощью класса `worktime` может составить отчеты о рабочем времени. Для правильного учета

рабочего времени необходимо учитывать только рабочие дни, исключая выходные и праздники. За расчет рабочих дней отвечает класс `working_day`, который возвращает количество рабочих дней за указанный период.

Диаграмма последовательности - это такая диаграмма, где на единой временной оси для определенного набора объектов изображен жизненный цикл какого-то конкретного объекта, а также взаимодействие действующих лиц (актеров) информационной системы в границах какого-то конкретного прецедента.

Элементы этой диаграммы включают в себя следующее: прямоугольники с наименованиями объектов, вертикальные линии времени для отображения хронологии, прямоугольники на пунктирных линиях времени, которые представляют деятельность объекта или выполнение им определенных функций, а также стрелки, указывающие на обмен сообщениями или сигналами между объектами.

При пересечении сотрудником контрольно-пропускного пункта с помощью Python приложения и веб-камеры распознается лицо и в базу данных заносится информация о том, во сколько сотрудник пришел на работу и ушел с нее.

При помощи веб-приложения, администратор, отвечающий за учет рабочего времени сотрудников, после авторизации может составлять отчеты о рабочем времени с использованием параметров выбора диапазона дат и фамилии сотрудника. В отчете отображается дата, фамилия, имя, отчество сотрудника, отдел, должность, а также время в которое сотрудник пришел и ушел, общее время работы, время переработки. Все отчеты возможно экспортировать в MS Excel (рисунки 14-16).

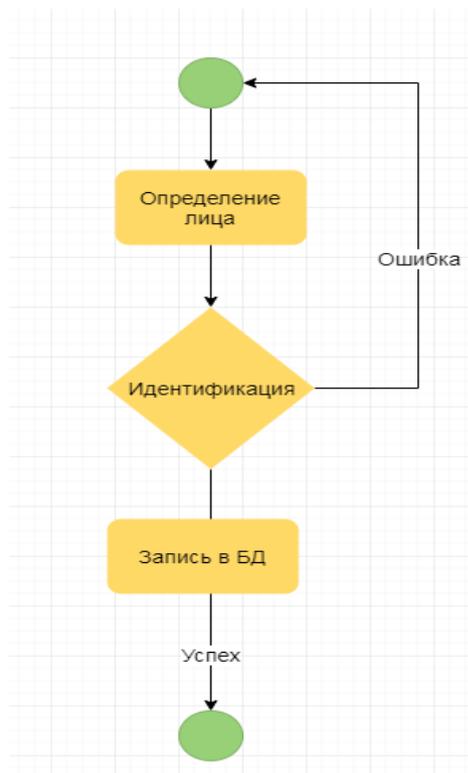


Рисунок 14 – Диаграмма деятельности для прецедента «Распознавание сотрудника»

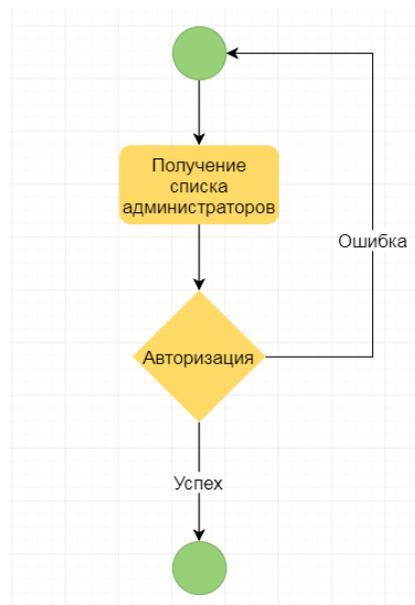


Рисунок 15 – Диаграмма деятельности для прецедента «Вход в систему»



Рисунок 16 – Диаграмма деятельности для прецедента «Формирование отчета»

С помощью указанных диаграмм визуально понятны функции и деятельность информационной системы. Как сам процесс регистрации рабочего времени сотрудника, так и процессы администрирования и управления рабочим временем персонала.

2.3 Техническое обеспечение

«Клиент-серверная архитектура представляет собой вычислительную модель, в которой сервер размещает, распределяет и контролирует большинство ресурсов, а также сервисы, которые будут использоваться клиентом. Такие структурные конструкции состоят из одной или нескольких клиентских систем, подключенных к центральным или основным серверам через сеть Интернет. Все системы, связанные с ним, совместно используют вычислительные ресурсы.

Архитектура клиент-сервер также называется сетевой вычислительной структурой, поскольку каждый запрос и связанные с ним сервисы распределены по сети. В архитектуре клиент-сервер, когда клиентский компьютер отправляет запрос данных на сервер через Интернет, сервер принимает запрос, обрабатывает его и доставляет запрошенные пакеты данных обратно клиенту. Особенность заключается в том, что серверный компьютер может одновременно управлять многочисленными клиентами. Кроме того, один клиент может одновременно подключаться к нескольким серверам, где каждый сервер предоставляет свой набор сервисов для этого конкретного клиента» [11].

Разделение задач между клиентами и сервером позволяет более эффективно использовать ресурсы информационных систем. Сервера, как правило, более мощные и надежные устройства, способные обрабатывать большие объемы данных, в то время как клиенты могут быть менее мощными устройствами (рисунок 17).



Рисунок 17 – Схема клиент-серверной архитектуры

Основные плюсы клиент-серверной архитектуры состоят в следующем:

- программы-клиенты не дублируют код программы-сервера;

- снижаются требования, предъявляемые к персональным компьютерам, на которых был установлен клиент, потому что на сервере осуществляются все вычисления;
- хранение всей информации осуществляется на сервере, а он защищён обычно намного лучше основной части клиентов. На сервере намного легче организовать контроль полномочий, для того чтобы исключительно клиентам с необходимыми правами доступа разрешать доступ к информации.

Недостатки:

- вся в целом вычислительная сеть может оказаться неработоспособной, при неработоспособности сервера. Неработоспособный сервер – это такой сервер, который находится на профилактике, ремонте и так далее, или сервер, у которого производительности не достаточно для обслуживания всех клиентов;
- поддержка функционирования этой системы требует системного администратора, то есть, отдельного специалиста.

В качестве веб-сервера в этой работе используется веб-фреймворк CherryPy.

CherryPy – это объектно-ориентированный фреймворк, написанный на языке программирования Python.

CherryPy является одним из старейших веб-фреймворков, доступных для Python, однако многие люди не знают о его существовании. Одна из причин состоит в следующем: CherryPy не является полным стеком со встроенной поддержкой многоуровневой архитектуры. Он не предоставляет интерфейсные утилиты и не предоставляет инструкций о том, как связываться с вашими базами данных. Вместо этого CherryPy позволяет принимать решения разработчику, не навязывая какие-либо решения. Это контрастная позиция по сравнению с другими известными фреймворками.

CherryPy имеет чистый интерфейс и делает все возможное, чтобы дать свободу выбора, в то же время предоставляя вам надежный фундамент для

построения приложения.

Несколько причин, по которым стоит выбрать CherryPy:

- простота (начать работать с CherryPy проще чем с другими фреймворками);
- эффективность;
- стабильность (полностью протестирован, оптимизирован и надежен для реального использования);
- большое сообщество;
- развертываемость (в отличие от других Python фреймворков существуют способы эффективной развертки);
- cherrypy полностью бесплатен.

CherryPy позволяет разработчикам создавать веб-приложения так же, как и любые другие объектно-ориентированные приложения на Python. Это приводит к уменьшению исходного кода и ускорению разработки.

Техническое обеспечение представляет собой совокупность средств передачи информации, компьютерной техники, технических средств, которые используются в информационных системах, а также в автоматизированных системах управления.

Комплекс технических средств этого проекта образуют:

- персональный компьютер (электронно-вычислительная машина);
- мышь-манипулятор;
- клавиатура;
- 2x веб-камера;
- подключение к сети компании.

В процессе выбора необходимого оборудования, необходимо исходить из следующих характеристик:

- надежность;
- производительность;
- ремонтпригодность;

– затраты.

Минимальные необходимые технические характеристики комплекса технических средств приведены в таблице 3.

Таблица 3 – Технические характеристики комплекса технических средств

Параметр	Характеристика
Процессор	AMD Ryzen 5 1600 OEM
Оперативная память	1,00 гб
Жесткий диск	150 гб
Операционная система	Linux
Видеокарта	Встроенная

Комплексы технических средств с подобными характеристиками не несут больших материальных затрат, легко заменяемы и высокоокупаемы. В данном случае считается оптимальным решением для внедрения информационной системы.

Выводы по главе 2

Во второй главе рассмотрена и описана логическая модель данных процесса учета рабочего времени сотрудников компании. Логическая модель представлена ER-диаграммой «сущность-связь». На основании логической модели данных была разработана датологическая модель базы данных и описаны атрибуты данных терминах СУБД.

Дано описание входной информации поступающей в систему и выходных отчетов формируемых программой.

Построена URL-диаграмма и диаграмма классов, описывающие процессы разрабатываемого ПО. Описана структура диалога и пользовательский интерфейс ПО учета рабочего времени сотрудников.

Дано описание технических средств необходимых для функционирования разрабатываемой системы.

Глава 3 Реализация и тестирование программного обеспечения ИС

3.1 Реализация программного обеспечения

Python - это популярный язык программирования, который широко применяется как для написания самостоятельных программ, так и для создания скриптов в различных областях. Он отличается высокой мощностью и переносимостью, что делает его привлекательным выбором для разработчиков. Простота в использовании и свободное распространение делают Python доступным и для новичков, и для опытных специалистов.

Python предоставляет обширный набор библиотек, что значительно упрощает реализацию различных проектов. Благодаря этим библиотекам, программисты могут эффективно работать над задачами разной сложности, начиная от маленьких скриптов до крупных и сложных проектов. Ориентация Python на высокую эффективность разработки и качество программного обеспечения делает его особенно привлекательным для использования как в стартапах, так и в больших корпоративных проектах.

Таким образом, Python представляет собой универсальный инструмент, который подходит для решения практически любых задач, благодаря своей гибкости, мощности и развитой экосистеме библиотек. [19]

Профессионально Python отлично подходит для бэкэнд-веб-разработки, анализа данных, искусственного интеллекта и научных вычислений. Многие разработчики также использовали Python для создания эффективных инструментов, игр и настольных приложений. Существует множество ресурсов, которые помогут вам ближе познакомиться с Python и решить ваши задачи.

Python обладает рядом значительных преимуществ, которые делают его одним из самых популярных языков программирования:

- простота в изучении и использовании: Python известен своей простотой и понятным синтаксисом, который делает его доступным

для новичков. Это позволяет быстро начать писать код и ускоряет процесс обучения программированию;

- мощная стандартная библиотека: Python поставляется с обширной стандартной библиотекой, которая включает модули и инструменты для решения разнообразных задач: от работы с файлами и сетевым взаимодействием до обработки данных и создания веб-приложений. Это упрощает разработку и позволяет быстрее достигать результатов;
- большое сообщество и поддержка: Python имеет активное сообщество разработчиков по всему миру. Это означает, что всегда есть множество ресурсов, форумов и библиотек, где можно найти помощь, советы и готовые решения для различных задач;
- мультиплатформенность: Python поддерживается на всех основных операционных системах (Windows, macOS, Linux), что делает его идеальным выбором для разработки кроссплатформенных приложений;
- широкие возможности для автоматизации: Python часто используется для написания скриптов и автоматизации задач, благодаря чему он становится востребованным инструментом в системном администрировании, DevOps и других областях;
- использование в различных областях: Python применяется практически во всех областях от разработки веб-приложений и научных исследований до анализа данных, искусственного интеллекта и машинного обучения. Это делает его универсальным языком программирования [41].

Проходя через турникет, сотруднику не надо прикладывать RFID, показывать удостоверение или выполнять какие-либо дополнительные действия. Для распознавания лиц автоматически используется приложение, состоящее из библиотеки TensorFlow и алгоритма FaceNet. TensorFlow представляет собой мощную платформу с открытым исходным кодом для

машинного обучения, которая обеспечивает разнообразные инструменты и библиотеки. Экосистема TensorFlow поддерживает широкий спектр задач машинного обучения, от базовых алгоритмов до передовых исследований. Благодаря этому ученым доступны самые передовые технологии, а разработчики могут легко создавать и развертывать приложения, основанные на машинном обучении.

Преимущества TensorFlow:

- простое построение моделей (TensorFlow предлагает несколько уровней абстракции, поэтому вы можете выбрать подходящий для ваших нужд);
- удобная разработка (TensorFlow позволяет легко обучать и развертывать вашу модель, независимо от того, какой язык или платформу вы используете);
- эффективное проведение исследований.

FaceNet представляет собой алгоритм распознавания лиц людей, который основан на TensorFlow. Изображение лица этот алгоритм преобразует в Евклидово пространство, в котором расстояние напрямую соответствует степени сходства лица. С применением триплетов осуществляется обучение. эффективность – это главное преимущество FaceNet. Возможность добиться точности распознавания лица составляет 99.63%.

Чтобы распознать лица работников, должны быть загружены фотографии каждого отдельного работника. Качество распознавания лиц работников будет тем выше, чем больше будет количество фотографий. Сбор значительного числа фотографий занял бы большое количество времени. В связи с этим было решено собрать от каждого работника по 1-му видео в формате mp4 и разбить это видео на множество фотографий.

Качественное распознавание сотрудников программой с различных ракурсов (при проходе через КПП человек может повернуться), нужно получить видео с лицом сотрудника с разных ракурсов вокруг его лица. В

данных целях был создан алгоритм записи видео для работников. [15]

Необходимо, чтобы запись видео осуществлялась на HD камеру, имеющую разрешение более 720p. В ходе записи видео работнику необходимо начать производить съемку с самой первой точки и камеру нужно перемещать последовательно по показанной на рисунке 20 траектории. Траектория записи видео представлена на рисунке 18 [17].

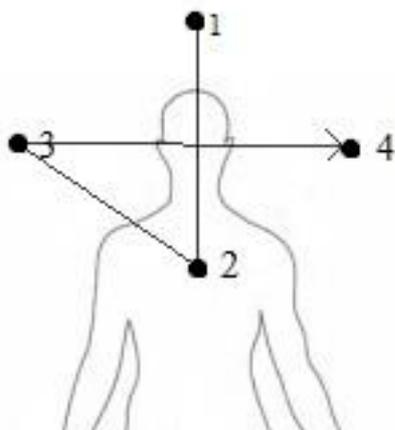


Рисунок 18 – Траектория записи видео

«Необходимо, чтобы длительность видео составляла от пятнадцати до двадцати секунд. Из данного видео может получиться двести пятьдесят – триста пятьдесят фотоснимков.

Видео всех работников, после их сбора, помещаются в паки, у которых название соответствует ID работника из БД (`emp_id`) и осуществляется обучение классификатора в целях последующего распознавания» [15].

Как только приложение распознает сотрудника на контрольно-пропускном пункте на четырех кадрах подряд, происходит запрос в базу данных к таблице `passing`, в которую вставляется четыре поля:

- `pass_id` (автозаполнение);
- `pass_post_id` (номер поста кпп);
- `pass_emp_id` (id сотрудника);

- `pass_time` (время прохождения через кпп).

В качестве веб-сервера используется фреймворк `CherryPy`, описанный выше. Для вывода данных с сервера на веб-страницу используется шаблонизатор `Jinja2`.

`Jinja2` – это современный и удобный язык шаблонов для `Python`, созданный по примеру шаблонов `Django`. Он быстрый, широко используемый и безопасный, с выполнением шаблона в изолированной среде.

Особенности:

- выполнение в песочнице;
- мощная автоматическая система экранирования `HTML` для предотвращения `XSS`;
- наследование шаблонов;
- компилируется до оптимального кода `Python`;
- заблаговременная компиляция шаблона;
- легко отлаживать (номера строк исключений напрямую указывают на строку с ошибкой строку в шаблоне);
- настраиваемый синтаксис.

`XSS` – это вид атаки на веб-системы, который состоит во внедрении вредоносного кода в выдаваемую веб-системой страницу (при открытии пользователем данной страницы он будет выполнен на его персональном компьютере) и в его взаимодействии с веб-сервером мошенника.

В целях работы с БД используется библиотека `SQLAlchemy`. `SQLAlchemy` – это набор инструментов для работы с `SQL` на языке `Python`, который предоставляет разработчикам приложений полную мощь и гибкость `SQL`. Строки могут быть выбраны не только из таблиц, но также из объединений и других операторов выбора; любые из этих значений могут быть составлены в большую структуру.

В таблице 4 представлено сравнение запроса к базе данных с помощью `SQLAlchemy` и простого `SQL` запроса.

Таблица 4 – Сравнение запроса SQLAlchemy и SQL запроса

SQLAlchemy	SQL
<pre>cherry.py.request.db['local'] .query(model.Employee) .outerjoin(model.Employee.department_id) .outerjoin(model.Employee.position_id) .order_by(model.Employee.id) .all()</pre>	<pre>SELECT emp_id, emp_name, emp_surname, emp_patron, emp_date_birth, emp_department, emp_posit FROM worktime.employee LEFT OUTER JOIN worktime.departaments ON worktime.departaments.dep_id = emp_department LEFT OUTER JOIN worktime.position ON worktime.position.pos_id = emp_posit ORDER BY emp_id</pre>

«Общий подход SQLAlchemy к задачам баз данных полностью отличается от подхода большинства других инструментов SQL. Вместо того, чтобы скрывать детали SQL и реляционные объекты за стеной автоматизации, все процессы полностью раскрываются в рамках серии компокуемых, прозрачных инструментов. Библиотека берет на себя задачу автоматизации избыточных задач, в то время как разработчик сохраняет контроль над тем, как организована база данных и как строится SQL» [20].

Для контроля качества и правильности кода используется Flake8. Flake8 – инструмент, позволяющий просканировать код проекта и обнаружить в нем стилистические ошибки и нарушения различных конвенций кода на Python. Использует соглашение PEP8.

PEP8 описывает соглашение о том, как писать код для языка Python, включая стандартную библиотеку, входящую в состав Python.

Программный код представлен в Приложении А.

3.2 Тестирование программного обеспечения

Ниже представлен пример оформления отчета с разбивкой по критичности дефектов.

Начинаем с наивысшего уровня критичности, переходя к минорным дефектам.

Таблица 5 – Отчет с разбивкой по критичности дефектов

Наименование	Последовательность действий	Категория дефекта	Критичность	Приоритет
Критичные дефекты				
Не работает обработка события отсутствия папок	Установить ПО без присутствия папок (promo, advertiseGoods, posm) в корне диска ПО автоматически запустилось после установки Фактический результат: Отображена маска загрузки и пустой экран приложения Ожидаемый результат: Сообщение об ошибке -нет соответствующих папок в корне диска (promo, advertiseGoods, posm)	Функциональность	Critical	Medium
Отсутствует название в меню при управлении приложением из трея	Скопировать файл *.json из папки json в папку promo Запустить приложение Нажать кнопку Пуск 4. Открыть управление приложением из трея Фактический результат: Пустое название в меню управления Ожидаемый результат: Название в меню содержит заголовок "Удалить из автозапуска"	Контент	Critical	Low

Продолжение таблицы 5

Наименование	Последовательность действий	Категория дефекта	Критичность	Приоритет
Блокирующие дефекты				
Отсутствует название в меню при управлении приложением из трея	Скопировать файл *.json из папки json в папку promo . Запустить приложение 3. Нажать кнопку Пуск 4. Открыть управление приложением из трея Фактический результат: Пустое название в меню управления Ожидаемый результат: Название в меню содержит заголовок "Удалить из автозапуска"	Контент	Critical	Low
Дефекты с медиум-приоритетом				
ПО не запускается на win7 32бит	Установить ПО на ОС windows 7 32 bit.(образ для oracle VM) Загрузка библиотеки .Net Установка ПО прошла успешно Фактический результат: ПО не запускается автоматически после установки Ожидаемый результат: ПО автоматически запускается после установки	Функциональность	Medium	Medium

Продолжение таблицы 5

Наименование	Последовательность действий	Категория дефекта	Критичность	Приоритет
Тривиальные дефекты				
Нет кнопки выхода из приложения	Запустить ПО. Выйти из ПО. Фактический результат: нет кнопки выхода из приложения Ожидаемый результат: на форме присутствует кнопка “Заккрыть”	Удобство	Minor	Trivial

Выводы по главе 3

В третьей главе приведено описание реализации и тестирования разрабатываемого программного обеспечения системы учета рабочего времени сотрудников ООО «И-Сети Телеком».

Языком программирования для написания ПО системы выбран язык Python.

Для распознавания лиц сотрудников применена библиотека с открытым кодом «TensorFlow».

Заключение

При выполнении данной работы исследовалась предметная область, проанализированы решения, которые существуют, и были сформулированы, исходя из этого, технические требования, предъявляемые к создаваемой системе.

Кроме того, осуществлен выбор комплекса средств разработки, в конечном счете, был смоделирован и был создан автоматизированный интерфейс учета РВ работников.

Были решены такие задачи данного исследования: по предметной области был собран и проанализирован материал, была обоснована необходимость создания системы автоматизации, а также была спроектирована структура этой будущей системы, для разработки этой информационной структуры был осуществлен выбор комплекса средств, была смоделирована ИС, и, вместе с тем, произведена программная реализация предложенного в работе проекта, а также осуществлена ее оценка.

В первой главе этой работы дано описание организационной структуры и экономической характеристики компании, была обоснована необходимость практической разработки автоматизированного интерфейса для учета РВ работников, а также для решения стоящей задачи были проанализированы существующие разработки.

Во второй главе этой работы обоснованы проектные решения по обеспечению (программному, технологическому, техническому, информационному) задачи, и, вместе с тем, был обоснован выбор программных средств.

Для разработки системы избранной архитектурой оказалась архитектура «клиент-сервер», в основе которой лежит Web-технология.

Было решено, что для создания ИС будет использован язык программирования Python с подключением фреймворка CherryPy, а для проектирования базы данных была использована реляционная система

управления базами данных MySQL.

Использование этой системы будет способствовать снижению трудоемкости.

Вместе с тем, использование этой системы даст возможность ликвидировать рутинные информационные операции.

У этой ИС есть интуитивно-понятный интерфейс, благодаря чему в работоспособности проекта работники могут разобраться без особого труда, в данной ИС могут формироваться отчеты, благодаря чему можно оперативно и своевременно вести учет РВ работников в организации.

В конечном счете, созданный программный продукт дает возможность выполнять любые задачи, которые нужны для эффективного учета РВ работников.

Вместе с тем, были приобретены при выполнении данной работы базовые, основные навыки проектирования ИС.

Список используемой литературы

1. АРМ «Орион про» [Электронный ресурс]: – Режим доступа: https://bolid.ru/production/orion/po-orion/po-arm/arm_orion_pro.html, свободный.
2. Бухгалтерский баланс ООО «И-Сеть Телеком» [Электронный ресурс]: – Режим доступа: <https://www.list-org.com/company/5487295>, свободный.
3. Грекул В. И., Денищенко Г.Н., Коровкина Н.Л. Проектирование информационных систем М.: Интуит НОУ, 2016. 123 с.
4. Ким Э. С. Анализ существующего программного обеспечения для автоматизации работы предприятия // Техника. Технологии. Инженерия. – 2019. – №1. – С. 11-14. – URL <https://moluch.ru/th/8/archive/36/857/>
5. Кудухов А. В. Разработка методов и алгоритмов интеллектуального кэширования информационных объектов в системах управления промышленными предприятиями М.: Интуит НОУ, 2010. 41 с.
6. Методы и средства проектирования [Электронный ресурс]: – Режим доступа: <http://elibrary.ru/item.asp?id=54032915>, ограниченный.
7. Модели данных и модели базы данных [Электронный ресурс]: – Режим доступа: https://studme.org/93786/informatika/modeli_dannyh_modeli_bazy_dannyh, свободный.
8. Нефункциональные требования: как не пустить систему ко дну [Электронный ресурс]: – Режим доступа: <https://habr.com/ru/companies/simbirsoft/articles/688428/>, свободный.
9. Организационная структура ООО «И-Сеть Телеком» [Электронный ресурс]: – Режим доступа: <https://k-telecom.org/>, ограниченный.
10. Особенности современного рынка систем контроля и управления доступом в России данных [Электронный ресурс]: – Режим доступа: <https://elibrary.ru/item.asp?id=29433884>, ограниченный.

11. Становление клиент-серверной архитектуры [Электронный ресурс]: – <https://elibrary.ru/item.asp?id=43162147>, ограниченный.
12. Сущности и отношения. Диаграммы классов и прецедентов [Электронный ресурс]. – Режим доступа: <https://e-educ.ru/bd15.html>, свободный
13. Устав Общества с ограниченной ответственностью «И-Сеть Телеком»
14. Фреймворки в веб-разработке [Электронный ресурс]. – Режим доступа: https://web-creator.ru/articles/about_frameworks, свободный.
15. Шайбаков И. А. Принципы автоматизации рабочего времени сотрудников телекоммуникационной компании М: Уральский государственный экономический университет, 2020 10-15 с.
16. Current UML specification [Электронный ресурс]. – Режим доступа: <https://www.omg.org/spec/UML/2.5.1/PDF>, свободный
17. Florian Schroff, Dmitry Kalenichenko, James Philbin "FaceNet: A Unified Embedding for Face Recognition and Clustering" – The Computer Vision Foundation, 2023 – 5 – 7 с.
18. Microsoft Application Architecture Guide, 2nd Edition [Электронный ресурс]. – Режим доступа: [https://learn.microsoft.com/ru-ru/previous-versions/msp-n-p/ff650706\(v=pandp.10\)](https://learn.microsoft.com/ru-ru/previous-versions/msp-n-p/ff650706(v=pandp.10)), свободный
19. Ramalho Luciano. Fluent Python. – O'Reilly Media, 2022 – 35 – 38 с.
20. Rick Copeland, Essential SQLAlchemy. – O'Reilly, 2008 – 5 с.

Приложение А

Фрагмент программного кода

```
import tkinter as tk
from tkinter.filedialog import askopenfilename, asksaveasfilename

class WorkTimeApp(tk.Tk):
    def __init__(self):
        super().__init__()
        self.title("Рабочее время")
        self.geometry('900x600')
        self.wm_geometry("+100+10")
        self.resizable(width=False, height=False)

        self.txt1 = self.create_text_widget(10, 30, 450, 400)
        self.txt2 = self.create_text_widget(10, 480, 130, 100)
        self.txt3 = self.create_text_widget(465, 30, 350, 400)
        self.create_buttons()
        self.create_labels()

        self.Tn = 1.0
        self.Tp = 1.0
        self.Totr = 0.0
        self.Kp = 0.0
        self.S1 = []
        self.Pi = 0.0
        self.result1 = []

    def create_text_widget(self, x, y, width, height):
        txt = tk.Text(self, font=12)
```

Продолжение Приложения А

```
txt.place(x=x, y=y, anchor="nw", width=width, height=height)
```

```
    txt.insert(1.0, "")
```

```
    return txt
```

```
def create_buttons(self):
```

```
    tk.Button(self, text="Загрузить", command=self.down1, bg="white",  
fg="black", font=("Helvetica Neue", 10)).place(x=370, y=433, anchor="nw",  
width=70, height=25)
```

```
    tk.Button(self, text="Загрузить", command=self.down2, bg="white",  
fg="black", font=("Helvetica Neue", 10)).place(x=150, y=550, anchor="nw",  
width=70, height=25)
```

```
    tk.Button(self, text="Вычислить", command=self.main1, bg="white",  
fg="black", font=("Helvetica Neue", 10)).place(x=735, y=433, anchor="nw",  
width=70, height=25)
```

```
    tk.Button(self, text="Сохранить", command=self.save, bg="white",  
fg="black", font=("Helvetica Neue", 10)).place(x=635, y=433, anchor="nw",  
width=70, height=25)
```

```
def create_labels(self):
```

```
    tk.Label(self, text="Сотрудник: отработ.время потер.время нагруз.месяц",  
font=("Helvetica Neue", 10), fg="blue").place(x=10, y=2, anchor="nw")
```

```
    tk.Label(self, text="Сотрудник: выплаты", font=("Helvetica Neue", 10),  
fg="blue").place(x=470, y=2, anchor="nw")
```

```
    tk.Label(self, text="Выплаты:", font=("Helvetica Neue", 10),  
fg="blue").place(x=10, y=450, anchor="nw")
```

```
def down1(self):
```

```
    op = askopenfilename()
```

```
    if op:
```

```
        with open(op, 'r') as f:
```

Продолжение Приложения А

```
content = f.read()
    self.txt1.delete(1.0, "end")
    self.txt1.insert(1.0, content)

def down2(self):
    op = askopenfilename()
    if op:
        with open(op, 'r') as f:
            content = f.read()
            self.txt2.delete(1.0, "end")
            self.txt2.insert(1.0, content)

def calculate(self):
    self.Kp = self.Tp / self.Tn
    content = self.txt2.get(1.0, "end").replace(" ", "").replace(",", ".")
    self.S1.clear()
    for i in content.split("\n"):
        if i:
            self.S1.append(float(i))
    self.Pi = (1 - self.Kp) * sum(self.S1)
    self.Pi = round(self.Pi, 2)
    self.result1.append(self.Pi)

def main1(self):
    self.result1.clear()
    content = self.txt1.get(1.0, "end")
    for line in content.split("\n"):
```

Продолжение Приложения А

```
self.Totr = 0.0
self.Tp = 1.0
self.Tn = 1.0
parts = line.split(":")
if parts:
    self.result1.append(parts[0])
    if len(parts) > 1:
        time_data = parts[1].split(" ")
        if len(time_data) > 0:
            self.Totr = float(time_data[0].replace(",", "."))
        if len(time_data) > 1:
            self.Tp = float(time_data[1].replace(",", "."))
        if len(time_data) > 2:
            self.Tn = float(time_data[2].replace(",", "."))
        if parts[1].strip():
            self.calculate()
self.update_results()

def update_results(self):
    self.txt3.delete(1.0, "end")
    str1 = ""
    for index, value in enumerate(self.result1):
        if index % 2 == 0:
            str1 += str(value) + " "
        else:
            if str(value) != "0.0":
                str1 += str(value) + "\n"
            else:
```

Продолжение приложения А

```
str1 += "\n"
    self.txt3.insert(1.0, str1)

def save(self):
    sa = asksaveasfilename()
    if sa:
        with open(sa + ".txt", "w") as file:
            for index, line in enumerate(self.result1):
                lm = str(line).replace("0.0", "").replace(".", ",")
                if index % 2 == 0:
                    file.write(lm + ' ')
                else:
                    file.write(lm + '\n')
if __name__ == "__main__":
    app = WorkTimeApp()
    app.mainloop()
```