

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)

01.03.02 Прикладная математика и информатика
(код и наименование направления подготовки / специальности)

Компьютерные технологии и математическое моделирование
(направленность (профиль)/специализация)

ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА (БАКАЛАВРСКАЯ РАБОТА)

на тему Моделирование системы оптимизации учета рабочего времени сотрудника

Обучающийся

Н. Н. Нечаев

(Инициалы Фамилия)

(личная подпись)

Руководитель

М. А. Тренина

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

Консультант

Е. В. Косс

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

Аннотация

Тема выпускной квалифицированной работы: «Моделирование системы оптимизации учета рабочего времени сотрудника».

Цель данной работы заключается в том, чтобы разработать модель системы оптимизации учета рабочего времени сотрудников.

Актуальность выпускной квалификационной работы заключается в оптимизации системы учета рабочего времени сотрудника, что позволит существенно уменьшить временные затраты. Сокращение нарушений трудовой деятельности сотрудников приведет к повышению результативности работы, путем минимизации числа прогулов сотрудниками рабочих часов. Моделирование системы оптимизации учета рабочего времени сотрудника необходима для практической реализации идеи эффективного управления деятельностью предприятия.

В первом разделе содержится сведения по теме, формулировании цели, задач, проблем, а также проведен обзор и сравнительный анализ часто используемых программных средств для учета рабочего времени.

Во втором разделе поставлена задача оптимизации учета рабочего времени, разработана модель системы оптимизации учета рабочего времени сотрудников и алгоритм работы.

В третьем разделе описывается реализация программного продукта для системы оптимизации учета рабочего времени.

Результатом работы является программный продукт, не имеющий недостатков существующих аналогов учета рабочего времени сотрудников.

ABSTRACT

The title of the graduation work is "Modeling the system for optimizing staff time recording".

The aim of the work is to develop a model of a system for optimizing staff time recording.

The relevance of the graduation work lies in optimizing the system for staff time recording, which will significantly reduce the time waste. Reducing work activities disruptions will lead to improvement in the performance by minimizing employee absenteeism. Modeling the system for staff time recording is necessary for the practical implementation of the idea of effective management at an enterprise.

The first chapter of the graduation work contains the formulation of goals, objectives, problems, as well as an overview and comparative analysis of commonly used software tools for staff time recording.

In the second chapter of the graduation work the task of optimizing the accounting of working hours is posed, a model of the system for optimizing the accounting of staff time recording and an algorithm of work are developed.

The third chapter of the graduation work describes the implementation of a software product for a staff time recording optimization system.

Overall, the result of the graduation work done is a software product that does not have the disadvantages of existing analogues for staff time recording.

Содержание

Введение	5
1 Обзор и сравнительный анализ систем учета рабочего времени	7
1.1 Цель и задачи систем учета рабочего времени в предприятии	7
1.2 Система учета рабочего времени DeskTime	8
1.3 Система учета рабочего времени CrocoTime	10
1.4 Система учета рабочего времени «СКУД и УРВ для 1С: Предприятие 8»	12
2 Моделирование системы оптимизации учета рабочего времени	16
2.1 Методика решения задачи оптимизации	16
2.2 Постановка задачи оптимизации системы учета рабочего времени сотрудников	19
2.3 Моделирование системы оптимизации учета рабочего времени	21
2.4 Выбор методики расчета потерь рабочего времени сотрудников предприятия	24
2.5 Алгоритм учета рабочего времени сотрудников	27
3 Реализация программного продукта для системы оптимизации учета рабочего времени	29
3.1 Реализация СКУД	29
3.2 Реализация реляционной базы данных Firebird	31
3.3 Реализация программного кода	38
Заключение	42
Список используемой литературы	43
Приложение А Фрагмент программного кода	45

Введение

Самая распространенная проблема, требующая оптимизацию метода управления персоналом, с которым сталкивается каждый предприниматель, является частое нарушение сотрудниками трудовой деятельности и для правильного решения вышеописанной проблемы существует учет рабочего времени сотрудников.

Для начала разберем основные понятия, связанных с учетом рабочего времени.

Каждый сотрудник, являющийся незаменимой частью всей огромной структуры предприятия, получает за выполнение своих обязанностей, поставленных начальством, заработную плату. Так как на выполнение этих обязанностей сотруднику требуется тратить свое личное время, то оно является рабочим. В ходе выполнения задач, для того чтобы предприниматель знал, как работают его сотрудники, собирается информация, которая является учетом рабочего времени.

Актуальность исследования состоит в том, что для более эффективного управления деятельностью предприятия следует вести учет рабочего времени сотрудников, чтобы сократить численность нарушений трудовой деятельности сотрудников, ведущих к временным затратам.

Объектом исследования является система оптимизации учета рабочего времени сотрудников.

Предметом исследования является модель системы оптимизации учета рабочего времени сотрудников.

Целью бакалаврской работы является разработка модели системы оптимизации учета рабочего времени сотрудников.

Для того, чтобы добиться выше поставленной цели, необходимо выполнить следующие задачи:

- провести обзор и сравнительный анализ известных ПО для учета рабочего времени сотрудника;

- изучить методы моделирования системы оптимизации учета рабочего времени;
- разработать модель системы оптимизации учета рабочего времени сотрудников;
- разработать алгоритм работы системы оптимизации учета рабочего времени;
- реализовать программный продукт для работы системы оптимизации учета рабочего времени.

Работа состоит из нескольких частей:

В первом разделе содержится сведения по теме, формулировании цели, задач, проблем, а также проведен обзор и сравнительный анализ часто используемых программных средств для учета рабочего времени.

Во втором разделе поставлена задача оптимизации учета рабочего времени, разработана модель системы оптимизации учета рабочего времени сотрудников и алгоритм работы.

В третьем разделе описывается реализация программного продукта для системы оптимизации учета рабочего времени.

В заключении описываются результаты выполнения выпускной квалификационной работы.

Приложение содержит фрагмент программного кода приложения.

Бакалаврская работа состоит из 50 страниц текста с приложением, 25 рисунков, 1 таблицы и 20 источников.

1 Обзор и сравнительный анализ систем учета рабочего времени

1.1 Цель и задачи систем учета рабочего времени в предприятии

Системы учета рабочего времени представляют собой мощнейший набор инструментов и имеют огромный потенциал для того, чтобы сотрудники выполняли свои обязанности в установленный руководством срок.

В основном, все сводится к тому, что происходит сбор максимально возможной информации о работе сотрудника и в ходе анализа ставится оценка эффективности выполнения сотрудниками своих задач, в ходе этих действий существенно снижаются временные затраты.

Для большинства предпринимателей проблемой может стать не только неэффективная работа сотрудников, а также недобросовестные сотрудники, которые имеют желание подработать нечестным способом, продавая конфиденциальную информацию. Не контролируя или игнорируя это, злоумышленники могут принести большой вред развитию предприятия.

Поэтому руководитель предприятия должен понимать, как работают его сотрудники, с чем ему на помощь приходит система учета рабочего времени для отслеживания и оценки эффективности сотрудников.

Подводя мини-итог, основная цель учета рабочего времени является сокращение до минимума временных затрат и сохранение конфиденциальной информации.

Задача учета рабочего времени состоит в том, чтобы фиксировать действия сотрудников, будь это приход на работу, обед, уход с работы, нахождение сотрудника на рабочем месте или же прогулы.

С помощью системы учета рабочего времени руководитель может следить за ходом выполнения работы сотрудников, а также управлять процессом работы.

1.2 Система учета рабочего времени DeskTime

Первая система, которая будет рассмотрена это DeskTime. Из особенностей можно отметить то, что это онлайн-сервис и для работы нужно просто установить приложение на устройство, что означает удобство и доступность для большинства начинающих предпринимателей. Данный сервис помогает руководителю автоматически отслеживать работу сотрудников в онлайн формате.

Основная цель DeskTime является повышение производительности и помощь командам и индивидуальным предпринимателям для того, чтобы они могли быстрее и более эффективней достичь желаемых результатов, путем автоматического отслеживания. Нельзя не согласиться, что этот сервис не даст больше времени, а лишь поможет грамотно его составить и использовать, что, собственно, уже является неплохим начальным шагом для дальнейшего развития.

На рисунке 1 представлено главное меню ПО DeskTime.

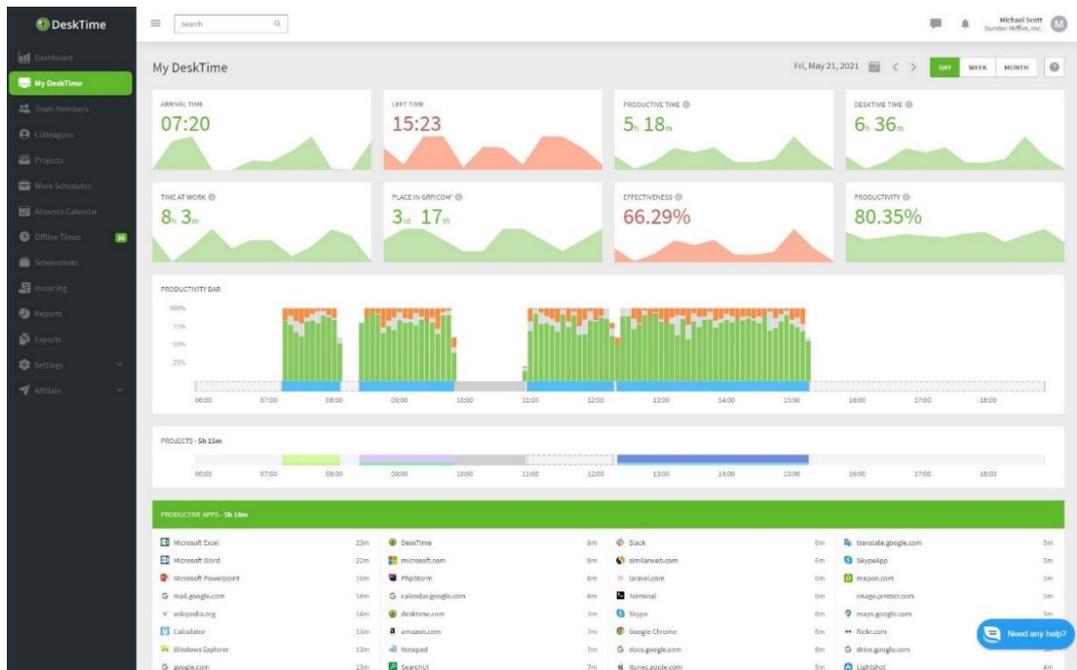


Рисунок 1 – Главное меню ПО DeskTime

Отслеживание времени – это довольно трудоемкий процесс. Важно следить и вести учет времени, которое требуется для решения тех или иных задач в течении фиксированного времени: день, неделя, месяц, год. В ходе развития предприятия это даже необходимо, являясь самым простым и удобным методом для учета времени. Следить за своим личным временем также важно для понимания учета времени. Важно замечать какие задачи требуют для решения больше времени, а какие, наоборот, требуют большего внимания.

Отметим такой ключевой фактор как автоматизация. Больше не нужно передавать информацию инструменту о выполненной работе, наоборот это инструмент будет сообщать о том, чем вы занимались в течении рабочего дня. Больше не требуется вводить данные самому вручную, для этого автоматизация и является преимуществом: можно избавиться от необходимости управлять таймерами, так как он точно записывает всю рабочую активность сотрудника в фоновом режиме.

Ниже, на рисунке 2, представлен перечень функций, выполнение которых обеспечивает программный продукт DeskTime.

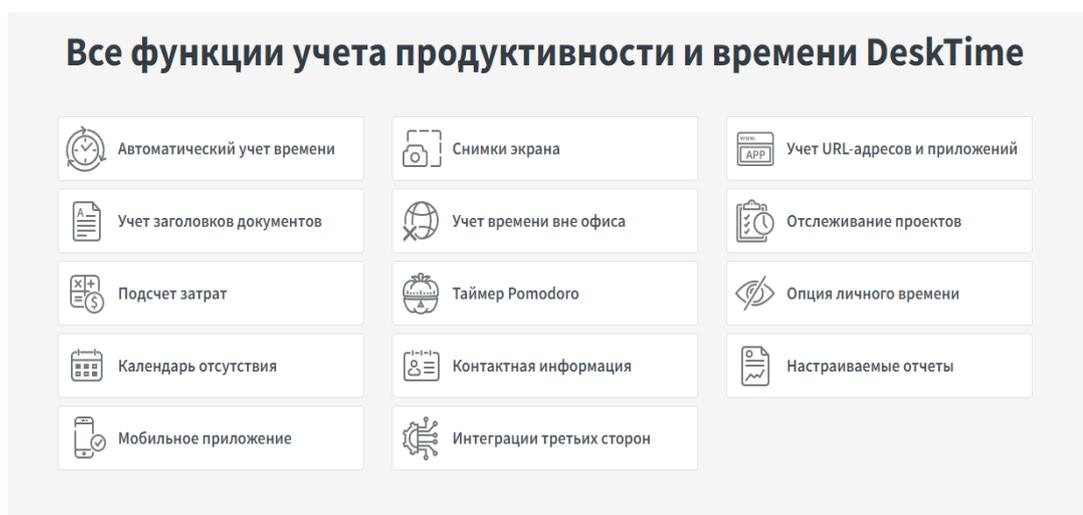


Рисунок 2 – Перечень функций, предоставляемых DeskTime

Главные задачи, решение которых предоставляет DeskTime:

– оказание помощи в нахождении малоэффективных подходов к работе.

Для того чтобы контролировать выполнение поставленных руководителем задач, с помощью функций DeskTime можно создавать отчеты о тех сотрудниках, которые внесли большой вклад в развитие предприятия или, наоборот, чьи подходы являются малоэффективными, и руководитель должен обратить свое внимание для изменения.

– учет рабочего времени для сотрудников, находящихся на дистанционном режиме работы. Если правильно контролировать работу сотрудников на дистанционке, то это может быть не только повысить эффективность работы, но и выйти дешевле для предприятия.

– функция свободного времени. По усмотрению руководителя, сотрудникам может выделяться свободное время, в эти моменты прекращается слежение за сотрудником, и он может потратить его на отдых.

– изменение рабочего времени. Во время командировок, отпусков, на выходных, сотрудник может временно приостановить учет рабочего времени [6].

1.3 Система учета рабочего времени CrocoTime

Вторая система, которая будет представлена называется CrocoTime. Как и выше рассмотренная система эта также является автоматической. Суть заключается в автоматическом отслеживании за сотрудниками, которые работают на своем рабочем устройстве.

На рисунке 3 изображено главное меню, с отчетами сотрудников отделов предприятия.

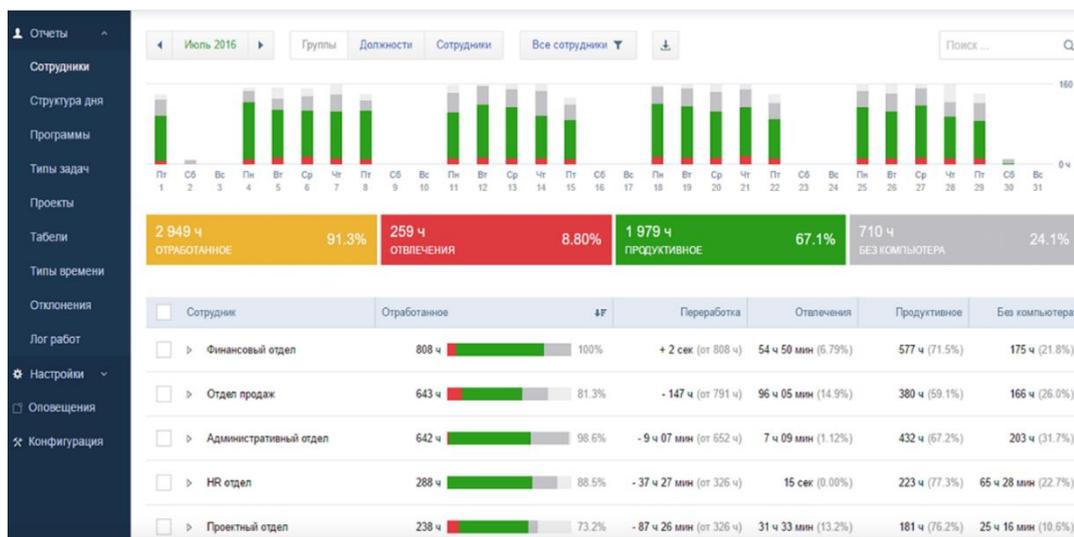


Рисунок 3 – Главное меню программного продукта CrocoTime, содержащее отчеты сотрудников отделов предприятия

Главные задачи, решение которых предоставляет CrocoTime:

- отслеживание активного времени. В ходе рабочего дня программа будет фиксировать действия сотрудника и в конце проанализирует результаты работы и формирует статистику по загруженным участкам времени в течении полного рабочего дня;

- фотоотчет. С помощью этой функции руководитель сможет следить за тем, чем занимался сотрудник, а также сколько времени он потратил на выполнение задания и какие программы для этого он использовал. Это очень поможет руководителям следить за теми сотрудниками, которые находятся дома на дистанционном режиме работы;

- анализ процессов. По желанию руководителя имеется возможность создать отчет по использованным программам и сайтам. Это очень удобно отслеживать для анализа бизнес-процессов предприятия;

- функция слежения за временем. Данная функция является неотъемлемой частью для учета рабочего времени. Необходимо для того, что подсчитать время, которое требуется на выполнения бизнес-проекта, определенные задачи и оценить работу, выполненную сотрудниками предприятия.

– табели учета. С помощью программы не обязательно ручное заполнение табелей учета рабочего времени сотрудников. Известные данные о графике работы, выходных, прогулов, больничных заполняются как вручную, так и в автоматическом режиме.

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

В конце для рассмотрения был выбран такой программный продукт, как «СКУД и УРВ для 1С: Предприятие 8». Эта программа, как и вышеперечисленные продукты, является средством для учета рабочего времени сотрудников на предприятии и считается ничем иным как внешней обработкой на базе технологической платформы «1С: Предприятие» версии 8.2 и 8.3.

В течении рабочего дня программа отслеживает сотрудников при входе на работу. То есть считывает приход и уход в здание, путем прохождения сотрудника предприятия через турникет. Все это происходит автоматически и в реальном времени.

Для того, чтобы понимать принцип работы системы учета рабочего времени, рассматриваемой в данном пункте, ниже, на рисунке 4, будет представлена общая схема работы.

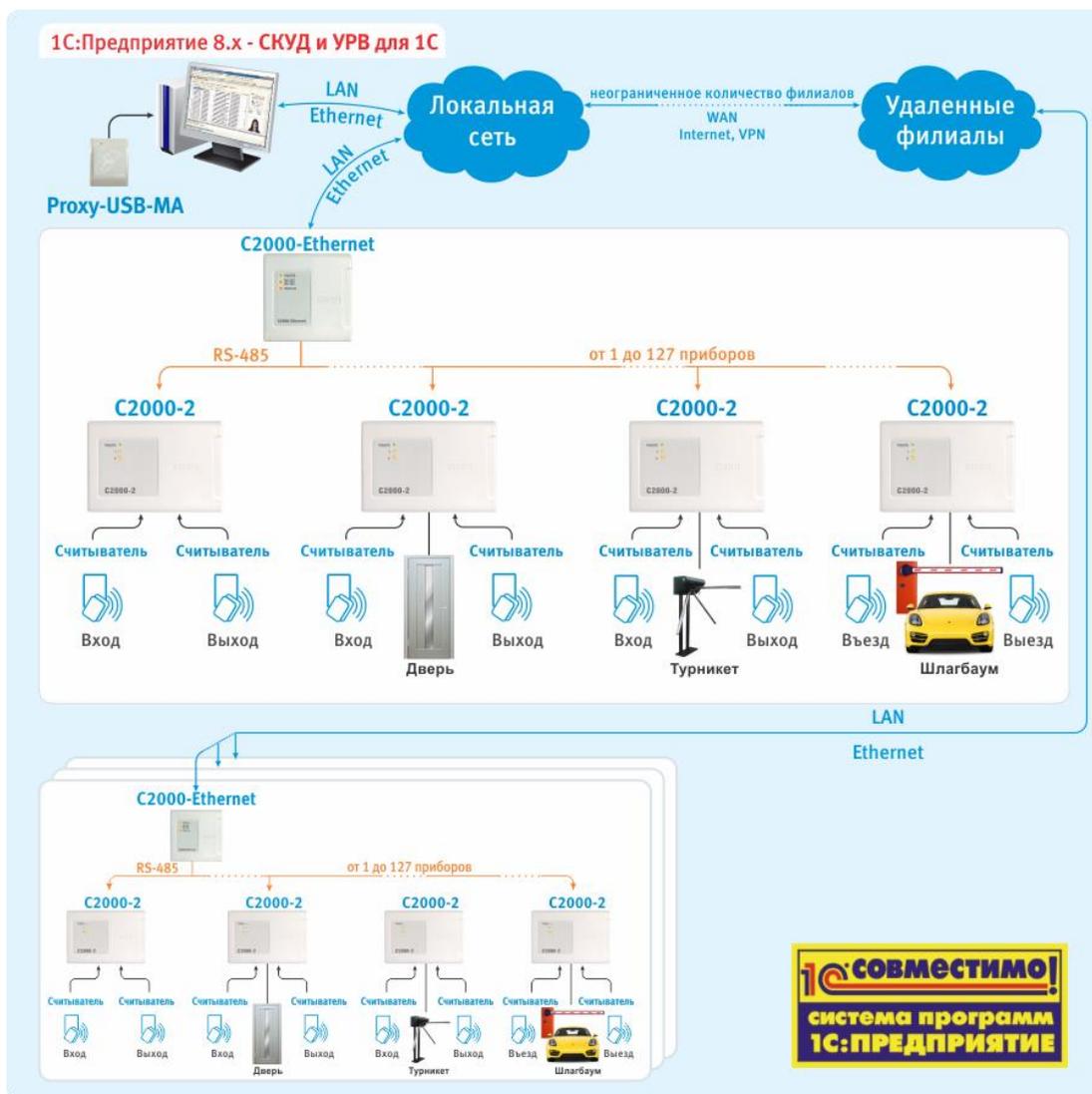


Рисунок 4 – Общая схема работы системы учета рабочего времени

В отличие от ранее рассмотренных систем учета рабочего времени данная программа имеет свои характерные плюсы и минусы, которые будут описаны далее. СКУД работает на базе технологической платформы «1С: Предприятие 8» и требуется предприятию для выполнения следующих функций:

- отслеживание прихода/ухода сотрудников в течении дня через двери или же турникетов;
- создание истории прихода/ухода сотрудников и сохраняет ее по дням;
- возможность выдачи сотрудникам уровень доступа или же наоборот лишение данных привилегий по желанию предпринимателя;

- запись ключей доступа в контроллеры;
- управлять доступом сотрудников по сценарию, написанному на языке 1С;
- выполнять учет рабочего времени по данным проходов сотрудников;
- автоматическое заполнение обязательного табеля учета рабочего времени данными о приходах / уходах сотрудников предприятия.

Из приведенного выше списка выполняемых продуктом функций можно сделать вывод, что с главной задачей всех подобных систем, а именно учет рабочего времени сотрудников, данное ПО справляется с обеспечением информационной поддержки предприятия [16].

Помимо этого, для применения данного ПО для отслеживания потребуется незначительное изменение для по причине разных особенностей учета и контроля рабочего времени сотрудников конкретного предприятия, а этом может вызвать как мелкие, так и большие денежные затраты для адаптации данной системы [9].

Для сравнения характеристик ПО, рассмотренных в этом разделе для учета рабочего времени сотрудников предприятия, рассмотрим таблицу 1.

Критерии оценивания:

- 0 – полное несоответствие требованиям;
- 1 – значительное несоответствие требованиям;
- 2 – для соответствия требованиям необходима полная переработка;
- 3 – для соответствия требованиям необходима незначительная доработка;
- 4 – незначительное несоответствие требованиям;
- 5 – полное соответствие требованиям.

Таблица 1 – Сравнительный анализ ПО для учета рабочего времени сотрудников предприятия

Характеристика ПО (1-5)	DeskTime	CrocoTime	СКУД и УРВ для 1С: Предприятие 8
стоимость покупки	2	2	1
сложность использования	3	3	3
сложность внедрения в предприятие	1	2	3
Итого	6	7	7

В ходе анализа можно прийти к выводу, что из перечисленных систем учета рабочего времени ни одна не соответствует установленным критериям.

Данный вывод в очередной раз показывает актуальность выбранной темы, и моделирование системы оптимизации учета рабочего времени поможет решить ставшую для предприятия проблему.

Выводы к разделу 1

Первый раздел в ВКР посвящена формулировке целей, задач, проблемам выбранной темы, обзорам и анализу существующих систем учета рабочего времени.

В ходе проделанной работы можно выделить такие выводы:

- системы учета рабочего времени необходимы предприятию для отслеживания рабочего времени сотрудников, что приводит к уменьшению временных затрат;

- необходимость внедрения систем учета рабочего времени состоит не только в уменьшении временных затрат, но и в ликвидации передачи конфиденциальной информации конкурирующим предприятиям / организациям;

- сравнение и анализ рассмотренных систем учета рабочего времени показал, что все они имеют свои недостатки, что дополнительно подчеркивает важность и актуальность выбранной темы.

2 Моделирование системы оптимизации учета рабочего времени

2.1 Методика решения задачи оптимизации

Основной задачей оптимизации является такой исход, при котором мы получаем наилучший для нас результат. Под наилучшим результатом мы понимаем такие критерии как: минимум затраченного времени, минимум денежных трат, получение максимальной прибыли и тому подобное. Проще говоря, задачей оптимизации является задача на отыскание оптимального решения.

Рассмотрим основные шаги для постановки задачи оптимизации.

Первым делом нужно знать и уметь переходить к математической постановке задачи, для этого требуется выполнить данные действия:

- 1) Необходимо определить целевую функцию

$$f(x): R^n \rightarrow R. \quad (1)$$

- 2) Определить множество допустимых решений $X \subset R^n$ для функции $f(x)$.

- 3) Выделить критерий оптимизации:

$$extr \in \{min, max\}. \quad (2)$$

Выполнив эти три пункта, мы получаем оптимизационную задачу. Ниже по формуле представлен вид математической постановки: $f(x) \rightarrow extr \ x \in X$ [2][3].

Главная суть данной задачи конечно же является в нахождении $x_0 \in X$, доставляющая min или max значение нашей ц.ф. (целевой функции) на множестве X из первого действия, то есть, другими словами, для x_0 выполняется одно из двух нижеперечисленных условий:

- 1) $f(x_0) \leq f(x)$ для всех x принадлежащих X ,

2) $f(x_0) \geq f(x)$ для всех x принадлежащих X .

Но представим ситуацию, что мы не нашли такой элемент, в виду его отсутствия, тогда придется построить последовательность, представленную в виде формулы под номером 3:

$$\{x_k\}, k = 1, 2, \dots, x_k \in X, \quad (3)$$

при этом должно выполняться одно из отношений 4 и 5:

$$\lim_{k \rightarrow \infty} f(x_k) = \inf_{x \in X} f(x), \quad (4)$$

$$\lim_{k \rightarrow \infty} f(x_k) = \sup_{x \in X} f(x). \quad (5)$$

Суть методов математической оптимизации заключается в том, чтобы найти максимально лучший результат без перебора всех других альтернативных методов, используя численные методы. Эти методы оптимизации делятся на условные и безусловные:

– задача безусловной оптимизации – это когда решение может находиться на всем множестве действительных чисел;

– задача условной оптимизации – это когда на область допустимых решений накладываются определенные ограничения, это называется областью допустимых решений.

В свою очередь методы безусловной оптимизации также имеет деление. Она исходит от размерности допустимого множества

– методы одномерной оптимизации;

– методы многомерной оптимизации.

Основную роль в ходе решения задач безусловной оптимизации выполняют численные методы поиска с нахождением экстремума и с помощью производных.

Эти методы можно разделить на три типа:

1) прямые методы, в которых целевая функция рассчитывается только в определенных точках;

2) методы первого порядка, в которых требуется вычислить первую частную производную;

3) методы второго порядка, в которых подразумевается вычисление второй частной производной. При отсутствии ограничений, а ц. ф. (целевая функция) является непрерывной, то такую задачу можно считать классической задачей на экстремум или же по-другому безусловный экстремум.

При наличии каких-либо ограничений для переменных в задаче оптимизации, то в решении лежит теория дифференциального исчисления, а именно, теория экстремумов и опирающиеся на нее методы.

Методы математического программирования классифицируются в зависимости от целевой функции и свойств допустимой области ограничений.

Пришло время рассмотреть основные признаки, определяющие то, что цель достигнута и модель является оптимизационной:

– первое и самое важное это то, что в модели имеется специальный признак. Это ничто иное как ц.ф. (целевая функция). Являясь самым важным признаком, ц.ф. определяет искомый критерий. Это может быть максимум дохода, эффективности работы, производительности, а также конечно же прибыли, то есть оптимизация в данных ситуациях должна быть максимальной. Такая же ситуация и с минимумом, целью является нахождение минимального числа издержек, трат денежных средств, потраченных сил, то есть, минимальный критерий;

– второе, обязательное и не менее важный признак – это система ограничений. Для того, чтобы получить максимально оптимальное решение нужно чтобы было из чего выбирать, поэтому решение должно быть четко ограничено (это могут быть сотрудники, сырье, качество оборудование и тому подобное). И немаловажно, чтобы эти ограничения вписывались в решение задачи, а именно чтобы каждый из них удовлетворяют условию задачи.

2.2 Постановка задачи оптимизации системы учета рабочего времени сотрудников

Основной причиной низких показателей при выполнении плана работы является нарушения трудовой деятельности сотрудниками. Этими нарушениями часто является отсутствие сотрудника на своем рабочем месте или занятие не рабочими делами, а своими собственными. Именно для предотвращения этих нарушений был придуман коэффициент потери рабочего времени, который помогает работодателю следить за эффективностью сотрудников.

Коэффициент рассчитывается по формуле ниже:

$$K_{\text{прв}} = T_{\text{прв}} / T_{\text{рнс}}, \quad (6)$$

где

- $T_{\text{прв}}$ – потери рабочего времени сотрудника в рабочее время за месяц;
- $T_{\text{рнс}}$ – рабочая нагрузка сотрудника в рабочее время за месяц [5].

Таким образом, задачу оптимизации рабочего времени сотрудника можно описать как задачу оптимизации вида:

$$K_{\text{прв}} \rightarrow \min, \quad (7)$$

с учетом ограничений продолжительности работ по ТК РФ [8].

Ранее описанный коэффициент в зависимости от сложившейся на предприятии ситуации может использоваться по-разному. В основном он используется в системе ключевых показателей эффективности (сокр. КПЭ) по мере значимости конкретных показателей [12].

Самый яркий пример таких ситуаций – это появление частых и постоянных прогулов сотрудников рабочих дней. В связи с этим, более разумно для руководителя при таких сотрудниках уменьшить показатель потери рабочего времени.

Тогда, когда данный показатель будет считаться выполненным, сотрудникам будет рассчитана премия по формуле:

$$P = (1 - K_{\text{прв}}) \times S_n, \quad (8)$$

где

- P – перерасчетные стимулирующие выплаты с учетом $K_{\text{прв}}$;
- S_n – сумма выплат денежных средств, рассчитанная для премии сотрудника [1].

Для того чтобы, выполнить решение данной задачи оптимизации реализована и представлена на рисунке 5 схема работы системы оптимизации учета рабочего времени, которая выполняет механизм учета рабочего времени сотрудников предприятия [11].

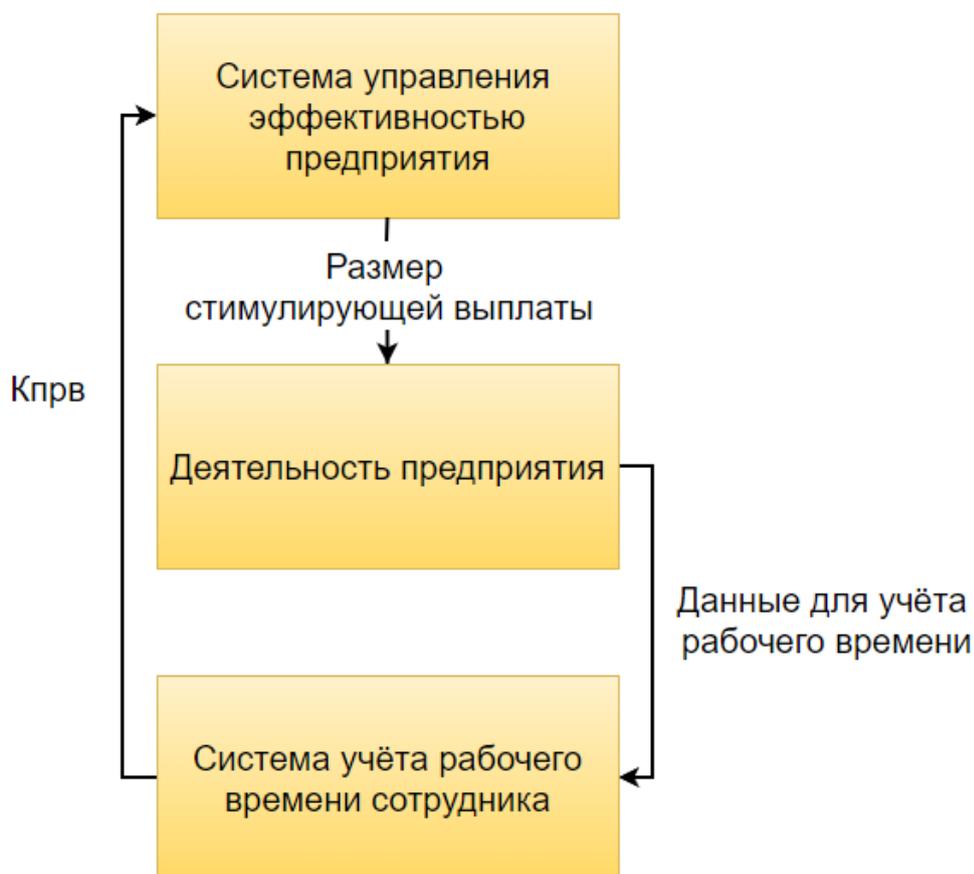


Рисунок 5 – Схема управления деятельностью сотрудников предприятия

Данные, которые будут использоваться в решении, являются кодами доступа сотрудников. С помощью них будет вестись учет рабочего времени.

Подводя итог, для реализации данного подхода, будет использован механизм учета рабочего времени сотрудников предприятия для более эффективного отслеживания деятельности сотрудников руководителем и последующих принятий решений в отношении сотрудника [10].

2.3 Моделирование системы оптимизации учета рабочего времени

Необходимость моделирования систем состоит в том, чтобы иметь представление об исследуемой системе, ее возможностях, сильных и слабых сторон, ее функциональность при работе.

В данном пункте рассмотрим концептуальное моделирование. В ходе моделирования концептуальной системы учета рабочего времени можно разработать несколько подходов для реализации, из них можно выделить две части, которые являются очень зависимыми по сравнению друг с другом:

- схема структуры. Схема структуры является совокупностью концепций об изучаемом объекте в определенной области;
- схема поведения. Если предыдущая схема изучала основные концепции, то схема поведения осуществляет реальные изменения, которые присущи самой системе.

Для реализации системы необходимо, чтобы она вмещала в себя всю информацию об изучаемом объекте, а также важно вмещать основные задачи, которая должна решать моделируемая система. Рассмотрим три основополагающие функции, которые должна осуществлять полученная программная система:

- функция памяти. Особенность данной функции является то, что она сохраняет знание о состоянии предметной области системы;

– информационная функция. Главная задача этой функции, в отличие от первой является информирование о состоянии предметной области системы;

– функция деятельности. Последняя функция, схожа со схемой поведения, осуществляет порядок действия, отвечающие за изменение состояния предметной области системы.

Говоря о состоянии предметной области, невозможно не упомянуть о существовании определенного ряда свойств. В тоже время само значение этих свойств зависит от конкретной цели, из-за которой и строилась сама система.

Предполагается, что рассматриваемая нами при реализации системы предметная область изучаемого объекта, является отношением понятий и объектов, из которых, собственно, и состоит система.

Анализируя методы моделирования концептуальной модели, стоит обратить внимание на то, что существуют определенного рода недостатки данных методов, а именно то, что они уже давно являются устаревшими.

Для представления структурно-функциональных моделей в большинстве случаев используют такие способы представления информации как известные нам таблицы, диаграммы, блок-схемы и тому подобное.

Для данной модели был выбран способ представления модели с помощью блок-схемы.

Коротко, блок-схема – это своего рода схема, выполненная с помощью шагов в виде различных фигур, отвечающих за определенные условия и действия, которая выполняет роль описания алгоритма.

Блок-схемы получили широкое применение в процессе разработки систем. Для ее создания в модели выбираются главные элементы, условия, действия.

Данный вид представления позволяет наглядно рассмотреть получившуюся систему, что является плюсом так как есть возможность найти причину неправильной работы алгоритма в глобальном масштабе.

К сожалению, не все так хорошо, как хотелось бы, проблема в том, что с помощью блок-схем невозможно уделить внимание всем мелочам, что не позволяет найти ошибку, которая закралась внутри алгоритма системы.

Блок-схема очень сильно востребована и полезна для представления входов и выходов системы, что побудило использовать в этом проекте для моделирования системы учета рабочего времени сотрудников предприятия [14].

Структурно-функциональная модель системы учета рабочего времени сотрудников предприятия в виде блок-диаграммы представлена на рисунке 6.

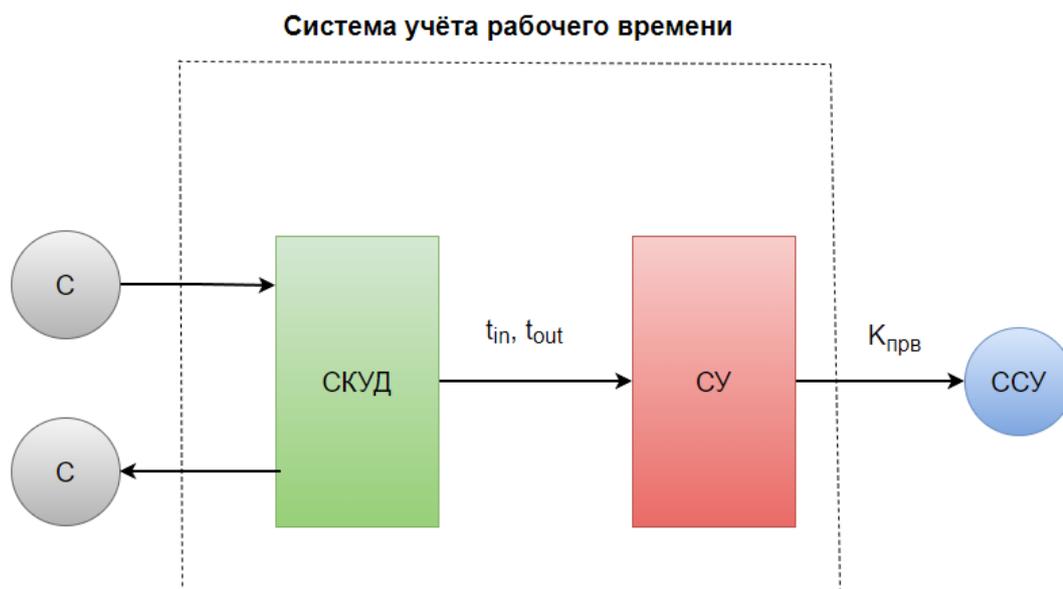


Рисунок 6 – Блок-диаграмма модели системы учета рабочего времени

Система учета рабочего времени состоит из нижеперечисленных пунктов:

– СКУД, собственно необходима для того, чтобы при входе/выходе сотрудников считывать информацию об этих событиях и передавать ее в систему управления.

В информацию о входе/выходе включаются:

t_{in} – время входа в здание предприятия;

t_{out} – аналогично, но только время выхода из здания предприятия.

– система управления – система, суть которой заключается в обработке полученной от СКУД информации о приходе/выходе сотрудников из здания через турникеты для сотрудников, следящих за учетом рабочего времени, что позволит качественно формировать для каждого сотрудника премиальные вознаграждения [4][13][15].

2.4 Выбор методики расчета потерь рабочего времени сотрудников предприятия

Выбору методики расчета рабочего времени сотрудников следует уделить серьезное внимание. Ведь методу необходимо разобрать и учесть множество факторов, например, вид рабочего графика, который является очень важной характеристикой, так как он может быть суточным, недельным, или переменным.

Рассмотрим 4 метода для учета рабочего времени сотрудников предприятия:

– метод «Все входы – выходы».

Фиксирует в течении дня все входы и выходы сотрудников. В методе не учитывается фактор обеда и перерывов, схематичное представление работы на рисунке 7.

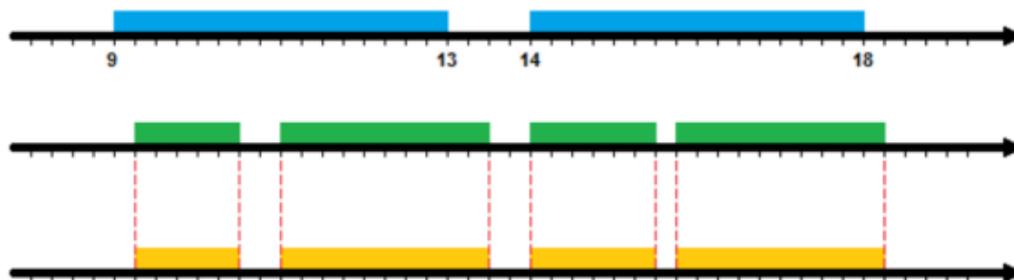


Рисунок 7 – Метод «Все входы - выходы»

Если руководителю необходимо учитывать только общее время присутствия сотрудника на рабочем месте, без учета перерывов, то специфика данного метода подойдет как никогда кстати.

– Метод «Первый вход – последний выход».

Данный метод, в отличие от первого рассмотренного уже более серьезно относится к обедам и перерывам. Система начинает фиксировать время с первого входа и заканчивает фиксировать в тот момент, когда сотрудник последний раз за день выходит.

– Метод перекуров.

Главной чертой данного метода является то, что при наличии в графике несколько промежутков, то их можно объединить в один, но следует учитывать тот факт, что, если разница между рассматриваемыми промежутками не более заранее определенного значения. Схематичное представление работы на рисунке 8.

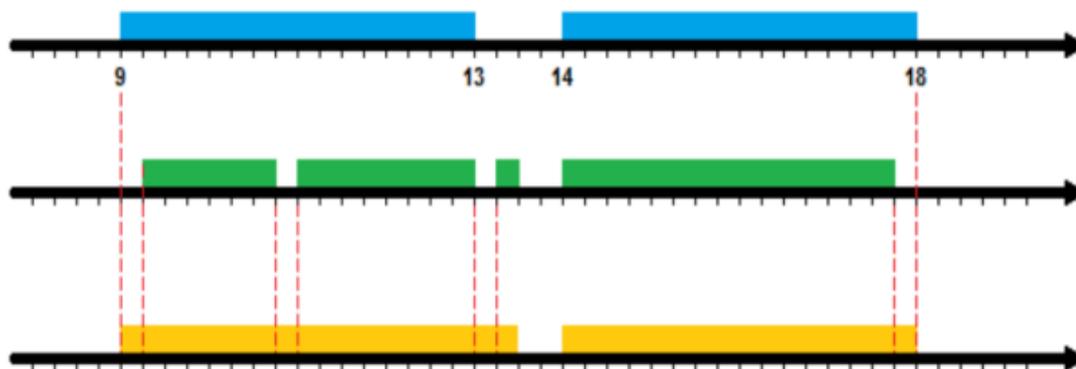


Рисунок 8 – Метод перекуров

Если график предприятия очень гибкий и устоявшегося графика работы нет, то данный метод им подойдет. Фиксированные промежутки времени можно объединить.

– Метод наложений.

Как и предыдущий метод, этот тоже может объединять промежутки рабочего времени, но с отличительной особенностью. Схематичное представление работы на рисунке 9.

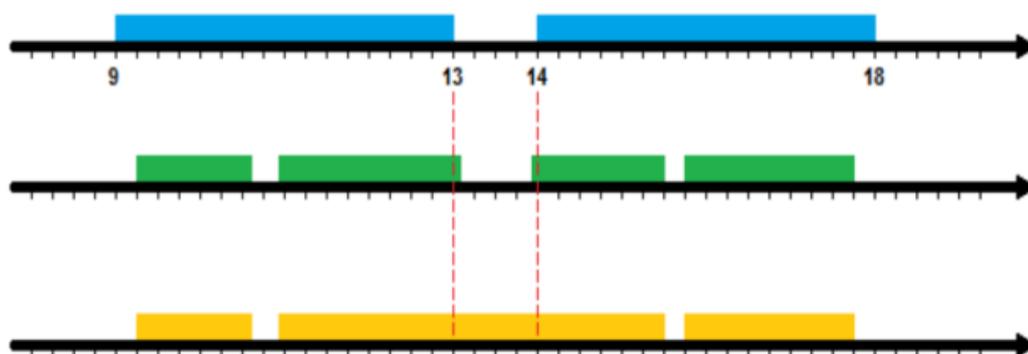


Рисунок 9 – Метод наложений

Таким образом, если сотрудник не работал в обед, то обеденный перерыв ему будет зачтен как рабочее время.

После рассмотрения предыдущих четырех методов для расчета потерь рабочего времени сотрудников предприятия был выбран метод «первый вход – последний выход» схематичное представление работы на рисунке 10.

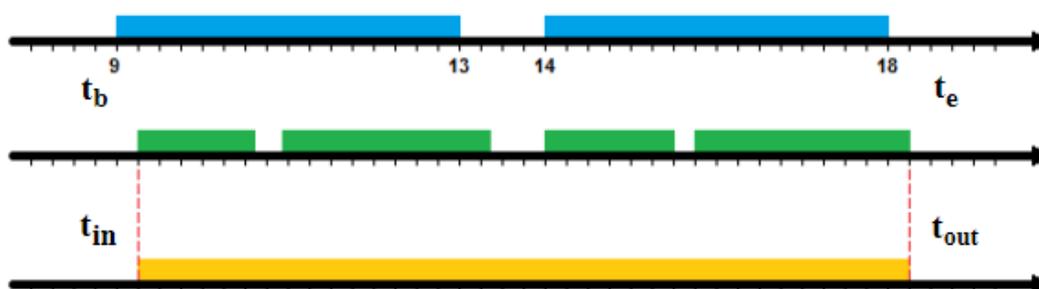


Рисунок 10 – Метод «первый вход- последний выход»

Из недостатков можно отметить тот факт, что данный метод является недостаточно точным для учета рабочего времени в условиях полного рабочего дня [19].

2.5 Алгоритм учета рабочего времени сотрудников

При входе/выходе сотрудника СКУД фиксирует моменты времени t_{in} и t_{out} .

Информация от СКУД направляется в систему управления.

Важно заметить, что при получении информации о входе/выходе используются только те, когда t_{in_i} и t_{out_i} , находятся в интервале (tb_i, te_i) , где tb_i и te_i – это начало или же конец i -го ($i = 1, 2, \dots, I$) рабочего дня определенного сотрудника.

На основе этого был разработан алгоритм учета рабочего времени сотрудника предприятия, которой имеет свойство «первый вход-последний выход».

Разберем шаги реализованного алгоритма:

Шаг 1. СКУД считывает время, когда сотрудник входит и выходит на территорию предприятия.

Шаг 2. Информация переданной информации о сотруднике передается в систему управления. В самой системе управления эти данные соотносятся с рабочим графиком сотрудника в этот день. Как было сказано ранее, используются только моменты времени t_{in_i} , t_{out_i} в интервале (tb_i, te_i) , где tb_i и te_i — это времена начала и конца i -го ($i = 1, 2, \dots, n$) рабочего расписания.

Шаг 3. Из формулы рассчитывается $K_{прв}$:

$$K_{прв}^{(n)} = \frac{\sum_i (B_{in_i}^{(n)} - t_{in_i}^{(n)}) + \sum_i (t_{out_i}^{(n)} - T_{out_i}^{(n)})}{T_{рнс}^{(n)}}, \quad (9)$$

где

– B_{in_i} , T_{out_i} – min и max значения t_{in_i} , t_{out_i} , соответственно,

– $T_{рнс}$ – нагрузка сотрудника.

Шаг 4. Полученный $K_{прв}$ необходим для расчета стимулирующих выплат сотруднику [20].

где

– n – является кодом доступа сотрудника в базе данных системы учета сотрудников;

– B_{in_i}, T_{out_i} – min и max значения t_{in_i}, t_{out_i} , соответственно.

Выводы к разделу 2

Второй раздел в ВКР посвящена описанию подходов к моделированию системы оптимизации учета рабочего времени, разработке модели системы оптимизации учета рабочего времени сотрудников предприятия.

В ходе проделанной работы можно выделить такие выводы.

Рассмотрена задача оптимизации учета рабочего времени вида:

$$\frac{\sum_i (B_{in_i}^{(n)} - t_{in_i}^{(n)}) + \sum_i (t_{out_i}^{(n)} - T_{out_i}^{(n)})}{T_{pnc}^{(n)}} \rightarrow \min.$$

Вследствие чего резко сократится число прогулов сотрудниками рабочих часов, уменьшающее потерю рабочего времени.

Построена модель системы оптимизации учета рабочего времени и разработана блок-диаграмма системы учета рабочего времени.

Рассмотрены методики подсчета учета рабочего времени и пришли к выводу, что оптимально выбрать метод «первый вход – последний выход».

Разработан алгоритм учета рабочего времени сотрудников. Цель которого является расчет специального коэффициента $K_{прв}$, который отвечает за выплату стимулирующих денежных средств каждого сотрудника.

3 Реализация программного продукта для системы оптимизации учета рабочего времени

3.1 Реализация СКУД

Основываясь на большой популярности использования APACS 3000 в предприятиях и высших учебных заведениях, в частности ТГУ, было принято решение использовать данный СКУД.

APACS 3000 содержит в себе множество функций для обеспечения корректной работы системы безопасности, а также данный программный комплекс имеет возможность контролировать доступ и вести учет рабочего времени.

Архитектура APACS 3000 представлена на рисунке 11.

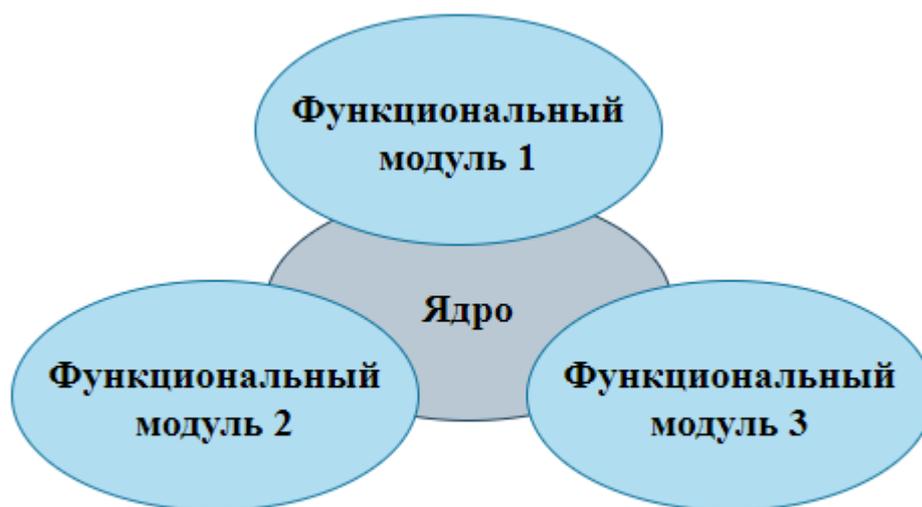


Рисунок 11 – Архитектура APACS 3000

Рассматривая основную структуру APACS 3000, можно выделить такие составные части: ядра системы и связанных с ним функциональные модули.

Задачи, выполнение которых осуществляет ядро:

– выполняет функцию сохранения целостности системы;

– является проводником для взаимодействия функциональных модулей системы;

– выполняет проверку имеющихся прав.

Функциональный модуль является неотъемлемой частью APACS 3000, ведь с помощью него появляется возможность добавить стороннее оборудование для более качественной работы системы.

APACS 3000 имеет возможность устанавливать свое ПО на несколько ПК, которые объединены одной сетью и каждый из этих ПК имеет право быть как в роли сервера APACS 3000, так и рабочей станцией.

Распределенная структура APACS 3000 представлена на рисунке 12.

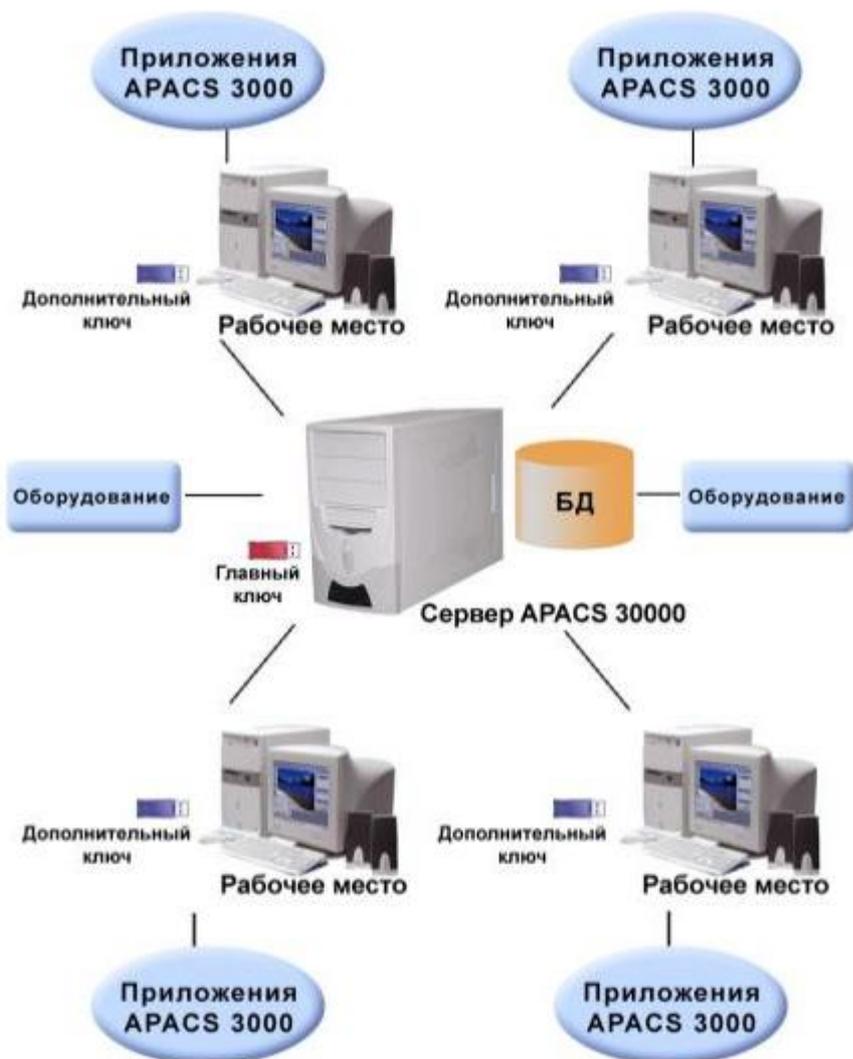


Рисунок 12 – Распределенная структура APACS 3000

Сервер APACS 3000 является ПК, на котором запущен сервер APACS 3000 и установлены дополнительное ПО, а также драйвера оборудования, обеспечивающие работу с локально подключенной аппаратурой [18].

Рабочие станции – являются рабочие места с устройством, в основном им является ПК, на которых установлены ПО APACS 3000 и другие клиентские приложения.

Система, состоящая из сервера APACS 3000 и нескольких рабочих станций, работающая с единой базой данных, является инсталлированной на объекте APACS 3000 [7].

3.2 Реализация реляционной базы данных Firebird

При входе или выходе сотрудников на территорию предприятия или здания, СКУД считывает информацию с электронного пропуска и передает ее в СУ, откуда и происходит одобрение на вход сотрудника и фиксирует время прибытия и убытия [17].

Для разработки базы данных на основе данных с турникета была выбрана бесплатная СУБД Firebird, которая интегрирована в APACS 3000.

Получаем данные входа и выхода персонала и на основе этих данных нужно рассчитать, кто опаздывает, на сколько, и на сколько уходит раньше.

Возьмем предположительное время работы с 8:00 до 17:00.

Создание объектов базы данных

```
CREATE TABLE Positions
```

```
(
```

```
id_posit      smallint NOT NULL,
```

```
posit        varchar(30) NOT NULL,
```

```
CONSTRAINT PK_Positions PRIMARY KEY(id_posit)
```

```
);
```

```
CREATE TABLE Deapartments
```

```

(
    id_depart      smallint NOT NULL,
    name_depart   varchar(30) NOT NULL,
    CONSTRAINT PK_Deapartments PRIMARY KEY(id_depart)
);

CREATE TABLE Employees
(
    id_emp         integer NOT NULL,
    lname         varchar(20) NOT NULL,
    fname         varchar(20) NOT NULL,
    mname         varchar(20) NOT NULL,
    id_posit      smallint NOT NULL,
    id_depart     smallint NOT NULL,
    CONSTRAINT PK_Employees PRIMARY KEY(id_emp),
    CONSTRAINT FK_1 FOREIGN KEY(id_posit) REFERENCES
Positions(id_posit),
    CONSTRAINT FK_2 FOREIGN KEY(id_depart) REFERENCES
Deapartments(id_depart)
);

CREATE TABLE Passage
(
    id_pass       integer NOT NULL,
    date_pass     date NOT NULL,
    time_pass     time NOT NULL,
    id_emp        integer NOT NULL,
    direct        char(1) NOT NULL,
    CONSTRAINT PK_Passage PRIMARY KEY(id_pass),

```

CONSTRAINT FK_3 FOREIGN KEY(id_emp) REFERENCES
Employees(id_emp)
);

Структура строения реализованной базы данных Firebird:

Структура таблицы Deapartments представлена на рисунке 13.

#	PK	BK	UNQ	Название	Тип	Домен	Длина	Точ...	Подтип	Массив	Не пусто	Кодировка	Коллате	Описани...
1	<input checked="" type="checkbox"/>			ID_DEPART	SMALLINT						<input checked="" type="checkbox"/>			
2				NAME_DEPART	VARCHAR		30				<input checked="" type="checkbox"/>	NONE	NONE	

Рисунок 13 – Структура таблицы Deapartments

Данная таблица содержит отделы предприятия.

Структура таблицы Employees представлена на рисунке 14.

#	PK	BK	UNQ	Название	Тип	Домен	Длина	Точ...	Подтип	Массив	Не пусто	Кодировка	Коллате	Описани...
1	<input checked="" type="checkbox"/>			ID_EMP	INTEGER						<input checked="" type="checkbox"/>			
2				LNAME	VARCHAR		20				<input checked="" type="checkbox"/>	NONE	NONE	
3				FNAME	VARCHAR		20				<input checked="" type="checkbox"/>	NONE	NONE	
4				MNAME	VARCHAR		20				<input checked="" type="checkbox"/>	NONE	NONE	
5				ID_POSIT	SMALLINT						<input checked="" type="checkbox"/>			
6				ID_DEPART	SMALLINT						<input checked="" type="checkbox"/>			

Рисунок 14 – Структура таблицы Employees

В данной таблице содержится информация о сотрудниках, а именно имя, фамилия, отчество, идентификационный номер сотрудника, а также идентификатор входа и выхода.

Структура таблицы Passage представлена на рисунке 15.

#	PK	BK	UNQ	Название	Тип	Домен	Длина	Точ...	Подтип	Массив	Не пусто	Кодировка	Коллате	Описани...	Вычислить как...	По умолчанию
1	<input checked="" type="checkbox"/>			ID_PASS	INTEGER						<input checked="" type="checkbox"/>					
2				DATE_PASS	DATE						<input checked="" type="checkbox"/>					
3				TIME_PASS	TIME						<input checked="" type="checkbox"/>					
4				ID_EMP	INTEGER						<input checked="" type="checkbox"/>					
5				DIRECT	CHAR		1				<input checked="" type="checkbox"/>	NONE	NONE			

Рисунок 15 – Структура таблицы Passage

Таблица Passage содержит информацию о входе и выходе сотрудников и фиксирует время их прибытия или отбытия.

Структура таблицы Positions представлена на рисунке 16.

#	PK	BK	UNQ	Название	Тип	Домен	Длина	Точ...	Подтип	Массив	Не пусто	Кодировка	Коллате
1	<input checked="" type="checkbox"/>			ID_POSIT	SMALLINT						<input checked="" type="checkbox"/>		
2				POSIT	VARCHAR		30				<input checked="" type="checkbox"/>	NONE	NONE

Рисунок 16 – Структура таблицы Positions

В данной таблице содержится информация о турникете, через который сотрудники проходят.

Для учета рабочего времени были выполнены запросы, позволяющие определить и рассчитать, кто опаздывает, на сколько, и на сколько уходит раньше.

Запрос: Кол-во отработанных часов по месяцам:

```
select lname, EXTRACT(MONTH FROM date_pass), SUM(quantity_hour)
from (select ((MAX(p.time_pass) - MIN(p.time_pass)))/3600 as
quantity_hour, p.date_pass, e.lname
from Passage p join Employees e ON p.id_emp = e.id_emp
GROUP BY p.date_pass, e.lname
)
GROUP BY lname, EXTRACT(MONTH FROM date_pass)
```

Результат выполнения запроса представлен на рисунке 17.

The screenshot shows a SQL Editor window with the following data:

LNAME	EXTRACT	SUM
Голубев	5	8,6666
Егорычева	5	18,0000
Маршалов	5	17,8333

Рисунок 17 – Результат выполнения запроса на количество отработанных часов по месяцам

Запрос: Список опоздавших за указанный день

```
select e.lname, p.time_pass
from Passage p join Employees e ON p.id_emp = e.id_emp
where p.time_pass > '8:00' AND p.date_pass = '05-01-2022' AND direct =
'П'
```

Результат выполнения запроса представлен на рисунке 18.

The screenshot shows a SQL Editor window with the following data:

LNAME	TIME_PASS
Голубев	08:15:00
Маршалов	08:10:00

Рисунок 18 – Результат выполнения запроса на список опоздавших за указанный день

Запрос: список сотрудников, которые ушли раньше окончания рабочего времени за указанный день

```
select e.lname, p.time_pass
from Passage p join Employees e ON p.id_emp = e.id_emp
where p.time_pass < '17:00' AND p.date_pass = '05-01-2022' AND direct =
'В'
```

Результат выполнения запроса представлен на рисунке 19.

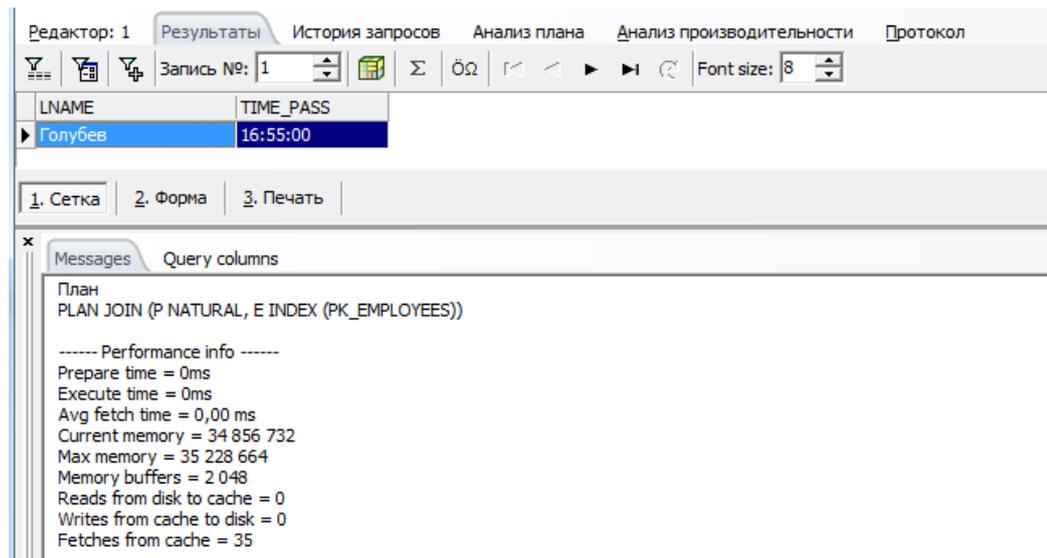


Рисунок 19 – Результат выполнения запроса на список сотрудников, которые ушли раньше окончания рабочего времени за указанный день

Запрос: список опоздавших за указанный период с указанием минут

```
select e.lname, (p.time_pass - time '8:00')/60, p.time_pass
from Passage p join Employees e ON p.id_emp = e.id_emp
where direct = 'п' AND (p.time_pass - time '8:00')/60 > 0
AND EXTRACT(MONTH FROM date_pass) = 5
ORDER BY (p.time_pass - time '8:00')/60 desc
```

Результат выполнения запроса представлен на рисунке 20.

LNAME	DIVIDE	TIME_PASS
Голубев	15,0000	08:15:00
Маршалов	10,0000	08:10:00
Егорычева	10,0000	08:10:00
Голубев	5,0000	08:05:00

1. Сетка | 2. Форма | 3. Печать

Messages Query columns

ПЛАН
PLAN SORT (JOIN (P NATURAL, E INDEX (PK_EMPLOYEES)))

----- Performance info -----
 Prepare time = 16ms
 Execute time = 0ms
 Avg fetch time = 0,00 ms
 Current memory = 34 860 036
 Max memory = 35 228 664
 Memory buffers = 2 048
 Reads from disk to cache = 0
 Writes from cache to disk = 0
 Fetches from cache = 47

Рисунок 20 – Результат выполнения запроса на список опоздавших за указанный период с указанием минут

Запрос: среднее время опозданий в минутах

```
select AVG(p.time_pass - time '8:00')/60
from Passage p join Employees e ON p.id_emp = e.id_emp
where direct = 'п' AND (p.time_pass - time '8:00')/60 > 0
```

Результат выполнения запроса представлен на рисунке 21.

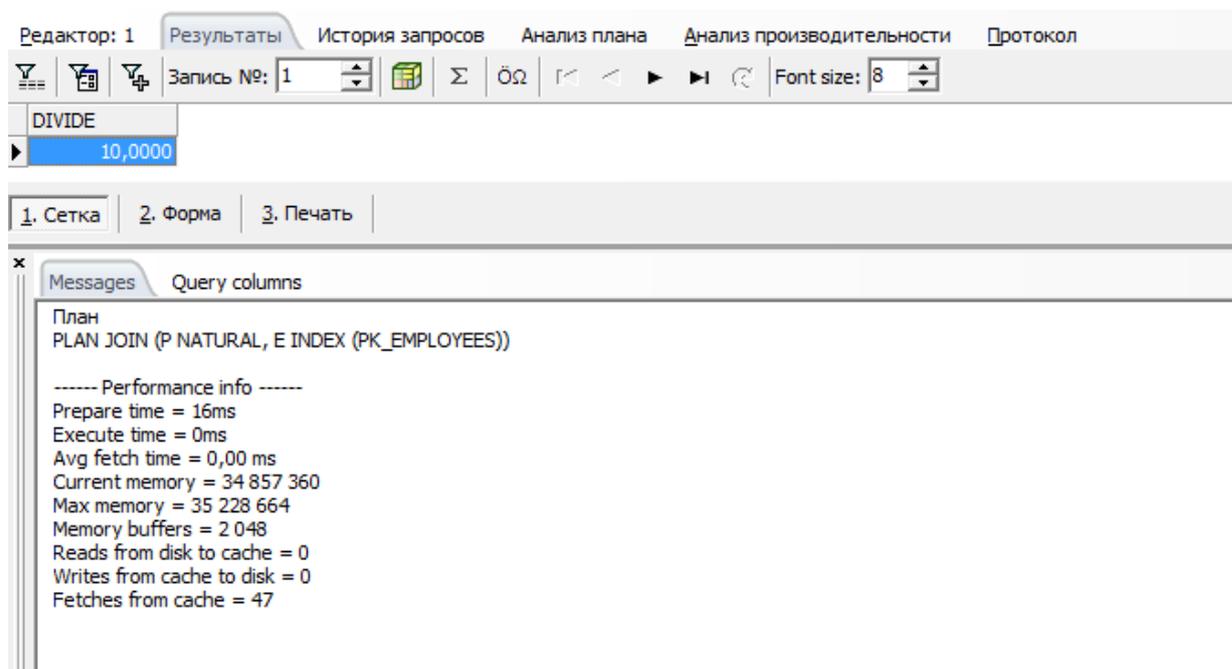


Рисунок 21 – Результат выполнения запроса среднее время опозданий в минутах

Реализованная база данных Firebird является неотъемлемой частью учета рабочего времени сотрудников.

Получая информацию с СКУД APACS 3000 о входе и выходе сотрудников из здания или выхода с территории предприятия, руководитель может отслеживать активность сотрудников, сокращая нарушения трудовой деятельности.

Также информация, полученная с СКУД, необходима для вычисления потерь рабочего времени с помощью метода «первый вход- последний выход» для дальнейшего нахождения коэффициента потерь рабочего времени и расчета стимулирующих выплат.

3.3 Реализация программного кода

Программный код написан на языке программирования Python. Данная программа считает рабочее время сотрудника за месяц и рассчитывает

коэффициент потерь рабочего времени и расчет стимулирующих выплат. Разработанный интерфейс программы представлен на рисунке 22.

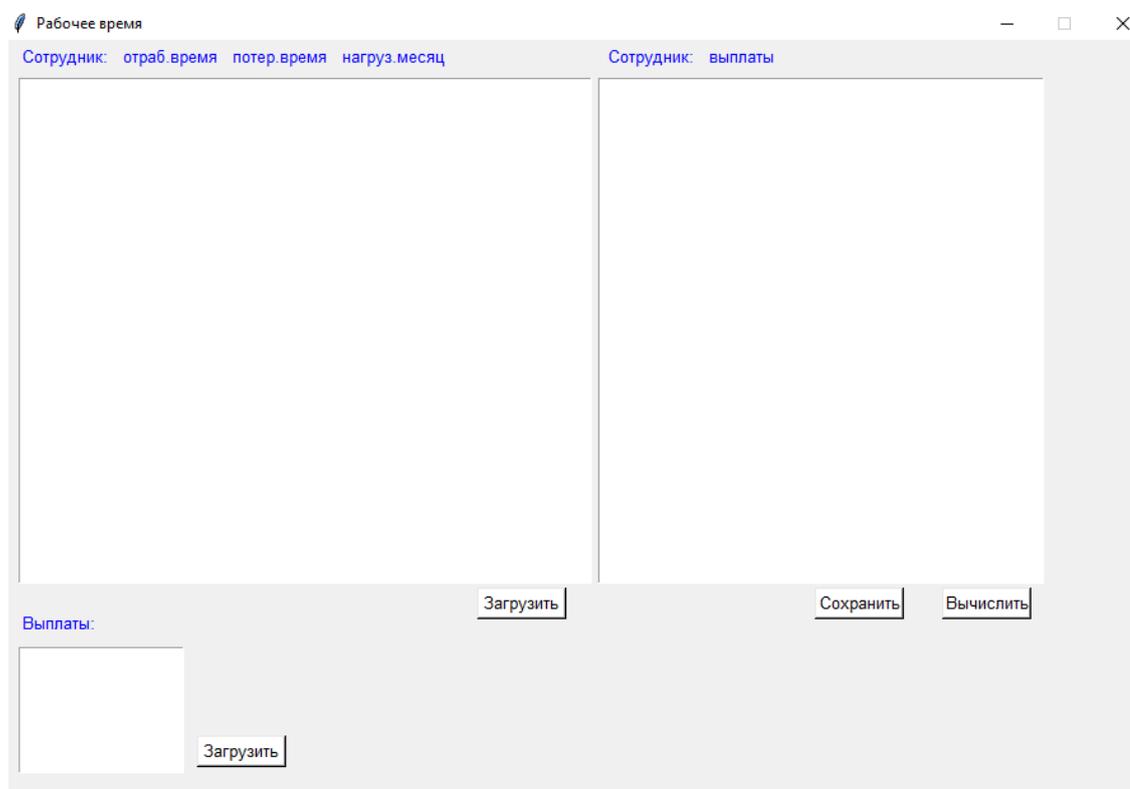


Рисунок 22 – Интерфейс программного кода для учета рабочего времени

Зафиксированная информация загружается в интерфейс. Она содержит уже полученную информацию о сотруднике: имя, фамилия, отдел, отработанное время, потерянное время и нагрузка сотрудника за месяц. Также загружается файл с выплатами для премий сотрудников. Стоит отметить, что данные могут быть загружены с файла, а также вписаны вручную для удобства.

На рисунке 23 представлен ввод изначальных данных в реализованную программу.

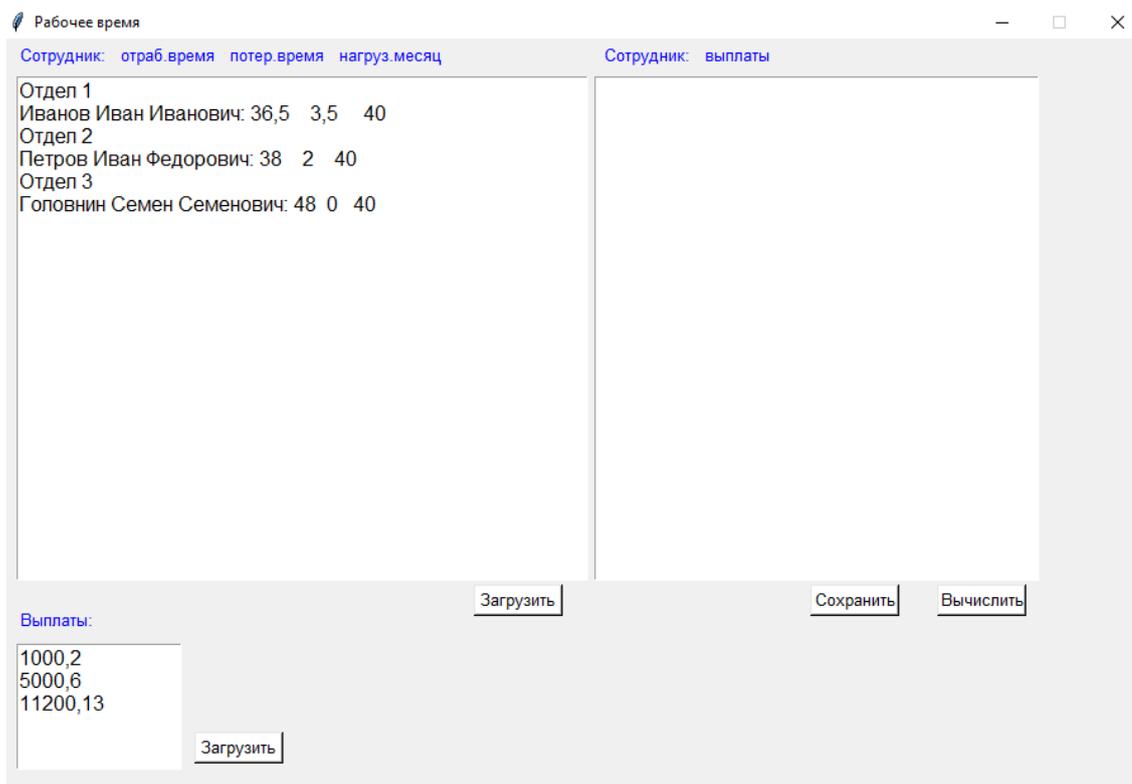


Рисунок 23 – Ввод изначальных данных в программу учета рабочего времени

Загрузив данные, вычисляется коэффициент потерь рабочего времени и перерасчетные стимулирующие выплаты.

На рисунке 24 представлен расчет данных.

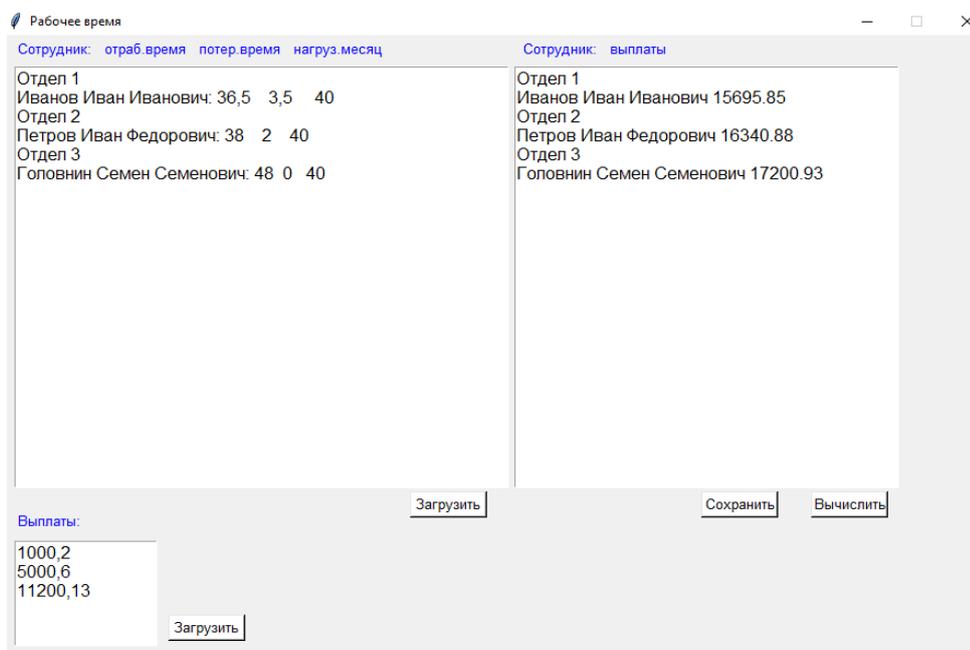


Рисунок 24 – Расчет коэффициента потерь рабочего времени и перерасчет стимулирующих выплат

```

def calculate():
    global Tn, Tp, Totr, Kp, S1, Pi, result1

    Kp = Tp / Tn # коэф потерь рабочего времени

    content = txt2.get(1.0, "end")
    content = content.replace(" ", "")
    content = content.replace(",", ".")
    S1.clear()
    for i in content.split("\n"):
        if i != "":
            print("i = ", i)
            S1.append(float(i))

    Pi = (1-Kp) * (sum(S1)) # выплаты денежных средств для премии по i-й гр показ
    Pi = round(Pi, 2)
    print(Pi)

    result1.append(Pi) |
    print(result1)

```

Рисунок 25 – Фрагмент программного кода расчета коэффициента потерь рабочего времени

Фрагмент программного кода представлен в Приложении А. Таким образом, реализован программный код для учета рабочего времени сотрудников предприятия.

Выводы к разделу 3

Третий раздел ВКР была посвящена разработке системы оптимизации учета рабочего времени.

В ходе проделанной работы можно сделать выводы:

- В качестве СКУД было решено использовать СКУД APACS 3000, широко используемая на предприятиях и высших учебных заведениях, в частности ТГУ.

- Для реализации системы управления было использована реляционная база данных Firebird.

- Реализовано соответствующее десктопное приложение на языке программирования Python.

Заключение

Основная задача системы оптимизации учета рабочего времени является улучшение деятельности сотрудников, путем уменьшения нарушений трудовой дисциплины, что повлияет на дальнейшее развитие и конкурентоспособность предприятия.

Для оценки эффективности деятельности сотрудников был рассмотрен коэффициент потери рабочего времени. На основе этого коэффициента была поставлена задача оптимизации учета рабочего времени для создания модели.

Главные результаты, полученные в ходе исследования:

– Проведен обзор и сравнительный анализ популярных систем учета рабочего времени сотрудников, в результате которого было проведено сравнение этих систем, результат которого показал, что существующие системы имеют существенные проблемы и тема исследования является актуальной.

– В ходе анализа и выбора подходов для моделирования системы оптимизации учета рабочего времени сотрудников было решено применить СКУД в качестве главной частью системы. При расчете потерь рабочего времени сотрудников используется метод «первый вход – последний выход».

– Разработаны модель и алгоритм работы системы оптимизации учета рабочего времени сотрудников. В процессе разработки модели использованы математическое моделирование, концептуальное моделирование, структурно-объектное моделирование.

– Для реализации системы оптимизации учета рабочего времени сотрудников было использовано: СКУД APACS 3000, реляционная база данных Firebird, а также реализовано соответствующее десктопное приложение на языке программирования Python.

Таким образом, в результате проделанной работы была частично решена актуальная в данный момент проблема путем моделирования системы оптимизации учета рабочего времени сотрудников.

Список используемой литературы

1. Бабешко, Л. О. Математическое моделирование финансовой деятельности. Учебное пособие / Л.О. Бабешко. - М.: КноРус, 2016. - 224 с.
2. Введение в математическое моделирование. Учебное пособие. - М.: Логос, 2015. - 440 с.
3. Галеев, Э. М. Оптимизация. Теория, примеры, задачи. Учебное пособие / Э.М. Галеев. - М.: Ленанд, 2015. - 344 с.
4. Павловский, Ю. Н. Компьютерное моделирование. Учебное пособие / Ю.Н. Павловский, Н.В. Белотелов, Ю.И. Бродский. - М.: Физматкнига, 2014. - 304 с.
5. Порядок расчета потери рабочего времени [Электронный ресурс]. URL: <http://rabotnik-info.ru/rabochee-vremya/poteri/>
6. Программа DeskTime [Электронный ресурс]. URL: https://deskttime.com/ru/programma-ucheta-rabochego-vremeni-sotrudnikov?dt_ads_source=google_search&gclid=Cj0KCQjwx7zzBRCcARIsABPRscPxMt7FNnXoqz-GKLTgfEX-zMuYVQAo4dqbu2D46rv6pAUMcuM9fzUaAvXxEALw_wcB
7. Программный комплекс APACS 3000 [Электронный ресурс]. URL: https://www.aamsystems.ru/programmnye_kompleksy/programmnyy_kompleks_apacs_3000/
8. Рейзлин, В. И. Математическое моделирование. Учебное пособие / В.И. Рейзлин. - М.: Юрайт, 2016. - 128 с.
9. СКУД: системы контроля и управления доступом [Электронный ресурс]. URL: <https://www.rmnt.ru/story/automation/skud-sistemy-kontrolja-i-upravlenija-dostupom.1522386/#go-celi-izadachi>
10. Федоткин, И. М. Математическое моделирование технологических процессов / И.М. Федоткин. - М.: Ленанд, 2015. - 416 с.
11. A. A. Samarskii and A. P. Mikhailov Principles of Mathematical Modeling Ideas, Methods, Examples published 2018 by CRC Press

12. Antić L. “Information support to motivation as phase of management process”, Facta Universitatis, Series: Economics and Organization Vol. 2, No 2, 2004. - P. 93 – 100.
13. Architecture of Access Control & Time Attendance System Solution [Электронный ресурс]. URL: <https://www.ravirajtech.com/architecture-of-access-control-system.html>
14. Block Diagram [Электронный ресурс]. URL: <https://www.smartdraw.com/block-diagram/>
15. Bosgra O.H., Kwakernaak H., and Meinsma G. “Design Methods for Control Systems”, Notes for a course of the Dutch Institute of Systems and Control Winter term, 2007–2008. – 321 p.
16. Bowers D.M. Access Control and Personal Identification Systems, Butterworth-Heinemann, 1988.
17. Eren C. CS 331-001: Database System Design and Management. – 2020. – URL: <https://docviewer.yandex.ru/view/311756418>
18. Models and Analysis in Distributed Systems / ed. by S. Haddad, F. Kordon, L. Pautet, L. Petrucci. – London-Hoboken: Wiley-ISTE, 2011. -368 p.
19. Time-tracking software [Электронный ресурс]. URL: https://en.wikipedia.org/wiki/Time-tracking_software
20. Vroom V. H. Work and motivation. New York: Wiley, 1964.

Приложение А

Фрагмент программного кода

```
root = Tk()
root.title("Рабочее время")
root.geometry('900x600') # размеры главного окна
root.configure()
root.wm_geometry("+%d+%d" % (100, 10)) # расположение главного окна
root.resizable(width=False, height=False)

txt1 = Text(font=12)
txt1.place(x=10, y=30, anchor="nw", width=450, height=400)
txt1.insert(1.0, "")

txt2 = Text(font=12)
txt2.place(x=10, y=480, anchor="nw", width=130, height=100)
txt2.insert(1.0, "")

txt3 = Text(font=12)
txt3.place(x=465, y=30, anchor="nw", width=350, height=400)
txt3.insert(1.0, "")

def down1():
    op = askopenfilename()
    with open(op, 'r') as f:
        content = f.read()

    txt1.delete(1.0, "end")
    txt1.insert(1.0, content)
```

Продолжение Приложения А

```
def down2():
```

```
    op = askopenfilename()
```

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

```
        content = f.read()
```

```
    txt2.delete(1.0, "end")
```

```
    txt2.insert(1.0, content)
```

```
def calculate():
```

```
    global Tn, Tp, Totr, Kp, S1, Pi, result1
```

```
     $Kp = Tp / Tn$  # коэф потерь рабочего времени
```

```
    content = txt2.get(1.0, "end")
```

```
    content = content.replace(" ", "")
```

```
    content = content.replace(",", ".")
```

```
    S1.clear()
```

```
    for i in content.split("\n"):
```

```
        if i != "":
```

```
            print("i = ", i)
```

```
            S1.append(float(i))
```

```
     $Pi = (1 - Kp) * (sum(S1))$  # выплаты денежных средств для премии по i-й гр
```

показ

```
    Pi = round(Pi, 2)
```

```
    print(Pi)
```

Продолжение Приложения А

```
result1.append(Pi)
print(result1)

def main1():
    global Tn, Tp, Totr, Kp, S1, Pi, result1
    result1.clear()
    content = txt1.get(1.0, "end")
    for i in content.split("\n"):
        Totr = 0.0
        Tp = 1.0
        Tn = 1.0
        print(i)
        n = 0
        for i2 in i.split(":"):
            if n == 0:
                result1.append(i2)
            if n == 1:
                print("----")
                n2 = 0
                for i3 in i2.split(" "):
                    if i3 != "":
                        i3 = i3.replace(",", ".")
                        if n2 == 0:
                            print("отраб время = ", i3)
                            Totr = float(i3)
                        if n2 == 1:
                            print("потери раб времени = ", i3)
                            Tp = float(i3)
```

Продолжение Приложения А

```
        if n2 == 2:
            print("нагрузка = ", i3)
            Tn = float(i3)
            n2 = n2 + 1
        print("----")
    n = 1
    if i != "":
        calculate()
print("все")

txt3.delete(1.0, "end")
str1 = ""
c1 = 0
for x1 in result1:
    if c1 == 0:
        str1 = str1 + str(x1) + " "
        c1 = 1
    else:
        if str(x1) != "0.0":
            str1 = str1 + str(x1) + "\n"
        else:
            str1 = str1 + "\n"
        c1 = 0
txt3.insert(1.0, str1)
```

Продолжение Приложения А

```
def save():
```

```
    global Tn, Tp, Totr, Kp, S1, Pi, result1
```

```
    sa = asksaveasfilename()
```

```
    with open(sa + ".txt", "w") as file:
```

```
        k1 = 0
```

```
        for line in result1:
```

```
            if k1 == 0:
```

```
                lm = str(line)
```

```
                lm = lm.replace("0.0", "")
```

```
                lm = lm.replace(".", ",")
```

```
                file.write(lm + ' ')
```

```
                k1 = 1
```

```
            else:
```

```
                lm = str(line)
```

```
                lm = lm.replace("0.0", "")
```

```
                lm = lm.replace(".", ",")
```

```
                file.write(lm + '\n')
```

```
                k1 = 0
```

```
btn = Button(root, text="Загрузить", command=down1, bg="white", fg="black",  
font=("Helvetica Neue", 10))
```

```
btn.place(x=370, y=433, anchor="nw", width=70, height=25)
```

```
btn = Button(root, text="Загрузить", command=down2, bg="white", fg="black",  
font=("Helvetica Neue", 10))
```

Продолжение Приложения А

```
btn.place(x=150, y=550, anchor="nw", width=70, height=25)
```

```
btn = Button(root, text="Вычислить", command=main1, bg="white", fg="black",  
font=("Helvetica Neue", 10))
```

```
btn.place(x=735, y=433, anchor="nw", width=70, height=25)
```

```
btn = Button(root, text="Сохранить", command=save, bg="white", fg="black",  
font=("Helvetica Neue", 10))
```

```
btn.place(x=635, y=433, anchor="nw", width=70, height=25)
```

```
lbl = Label(root, text="Сотрудник: отраб.время потер.время нагруз.месяц",  
font=("Helvetica Neue", 10), fg="blue")
```

```
lbl.place(x=10, y=2, anchor="nw")
```

```
lbl = Label(root, text="Сотрудник: выплаты", font=("Helvetica Neue", 10),  
fg="blue")
```

```
lbl.place(x=470, y=2, anchor="nw")
```

```
lbl = Label(root, text="Выплаты:", font=("Helvetica Neue", 10), fg="blue")
```

```
lbl.place(x=10, y=450, anchor="nw")
```

```
root.mainloop()
```