



## **Аннотация**

Тема: Разработка проекта автоматизированной системы учета тарификации преподавателей на примере ГАПОУ ТСПК Тольяттинского социально-педагогического колледжа.

Ключевые слова: АИС тарификации преподавателей, автоматизация бизнес-процессов, нагрузка, дисциплины, учебные группы.

Объектом исследования в данной работе является учет тарификации преподавателей.

Предметом исследования является процесс разработки проекта автоматизированной системы учета тарификации преподавателей.

Внедрение автоматизированной системы учета тарифицированных ставок преподавателей будет способствовать оптимизации управленческих процессов, снижению трудовых и финансовых издержек, а также повышению общей эффективности работы образовательного учреждения.

Проект автоматизированной системы будет адаптирован под конкретные потребности ГАПОУ ТСПК, включая особенности учебного плана и структуру преподавательского состава.

Объём дипломной работы 58 страниц, на которых размещены 44 рисунка и 8 таблиц. При написании диплома использовалось 20 источников.

## Оглавление

|   |    |
|---|----|
| Введение .....  | 6  |
| Глава 1 Функциональное моделирование предметной области .....   | 9  |
| 1.1 Техничко-экономическая характеристика предметной области .....  | 9  |
| 1.2 Концептуальное моделирование предметной области.....  | 11 |
| 1.2.1 Выбор технологии концептуального моделирования предметной<br>области .....  | 11 |
| 1.2.2 Моделирование бизнес-процессов предметной области для<br>постановки задачи автоматизированного варианта решения ..... | 13 |
| 1.2.3 Разработка и анализ модели бизнес-процесса «как есть» .....   | 13 |
| 1.2.4 Обоснование необходимости автоматизированного варианта<br>решения и формирование требований к новой технологии .....  | 15 |
| 1.3 Анализ существующих разработок на предмет соответствия<br>сформулированным требованиям.....                             | 16 |
| 1.3.1 Определение критериев анализа .....   | 16 |
| 1.3.2 Сравнительная характеристика существующих разработок .....  | 17 |
| 1.4 Постановка задачи на разработку проекта создания АИС .....  | 18 |
| 1.5 Разработка модели бизнес-процесса «как должно быть» .....   | 18 |
| Глава 2 Логическое проектирование АИС .....   | 22 |
| 2.1 Выбор технологии логического моделирования АИС .....  | 22 |
| 2.2 Логическая модель АИС и её описание .....   | 23 |
| 2.3 Информационное обеспечение АИС .....  | 23 |
| 2.3.1 Используемые классификаторы и системы кодирования .....   | 23 |
| 2.3.2 Характеристика нормативно-справочной и входной оперативной<br>информации .....  | 24 |
| 2.3.3 Характеристика выходной информации.....   | 24 |
| 2.4 Проектирование базы данных АИС .....  | 24 |
| 2.4.1 Выбор технологии проектирования БД АИС .....  | 24 |
| 2.4.2 Разработка концептуальной модели данных АИС .....   | 25 |

|  |    |
|--|----|
| 2.4.3 Обоснование вида логической модели .....                               | 25 |
| 2.4.4 Разработка логической модели данных АИС .....                          | 26 |
| 2.5 Требования к аппаратно-программному обеспечению АИС .....                | 26 |
| Глава 3 Физическое проектирование АИС .....                                  | 29 |
| 3.1 Выбор архитектуры АИС .....  | 29 |
| 3.2 Выбор технологии разработки программного обеспечения АИС .....           | 30 |
| 3.3 Выбор СУБД АИС .....   | 31 |
| 3.4 Разработка физической модели данных АИС .....                            | 32 |
| 3.5 Разработка программного обеспечения АИС .....                            | 34 |
| 3.5.1 Схема взаимосвязи модулей приложения АИС .....                         | 34 |
| 3.5.2 Описание модулей приложения АИС с примерами программного<br>кода ..... | 35 |
| 3.6 Описание функциональности АИС .....                                      | 36 |
| 3.7 Тестирование программного проекта .....                                  | 44 |
| 3.8 Обоснование экономической эффективности проекта .....                    | 48 |
| Заключение .....   | 54 |
| Список используемой литературы .....   | 56 |
| Приложение А Код программы .....   | 58 |

## Введение

Научно-технический прогресс позволил внедрить автоматизированные информационные системы практически во все сферы деятельности. Не осталось в стороне и образовательная система. Учебные заведения получили новые возможности для отслеживания образовательного процесса, информационные системы позволили собрать разрозненную информацию в единое целое. Системы мониторинга и анализа дали возможность определить проблемы в образовательной системе [2], [16].

Актуальность разработки проекта автоматизированной системы учета тарификации преподавателей на примере ГАПОУ ТСПК определяется нижеперечисленными ключевыми аспектами.

Ручной расчет заработной платы преподавателей часто сопровождается ошибками, что приводит к недовольству сотрудников и финансовым потерям учреждения. Автоматизированная система учета тарификаций позволяет исключить человеческий фактор при расчете оплаты труда, обеспечивая точность и прозрачность всех финансовых операций.

Процесс расчета зарплаты вручную занимает значительное количество времени у бухгалтерии и других ответственных лиц. Внедрение автоматизированной системы значительно сократит эти временные затраты, позволяя сотрудникам сосредоточиться на более важных стратегических задачах.

Автоматизированный подход обеспечивает единообразие расчетов независимо от количества преподавателей и специфики их нагрузки. Это снижает вероятность ошибок, связанных с различиями в методиках расчета, и способствует соблюдению единых стандартов учета.

Законодательство в сфере образования постоянно меняется, и требования к учету рабочего времени и оплате труда становятся все строже. Автоматическая система поможет своевременно учитывать изменения в законодательстве и поддерживать соответствие всем нормативам.

Система обеспечит удобный доступ к данным о тарификации каждого преподавателя, что повысит уровень доверия со стороны сотрудников и облегчит процесс контроля за расходованием средств. Преподаватели смогут самостоятельно отслеживать свои начисления, а руководство – оперативно получать аналитические отчеты.

Проект автоматизированной системы может быть адаптирован под конкретные потребности ГАПОУ ТСПК, включая особенности учебного плана, структуру преподавательского состава и другие важные аспекты. Это делает систему гибкой и легко масштабируемой в случае расширения или изменений в организации учебного процесса.

Ошибки в учете могут привести к штрафам и другим санкциям со стороны контролирующих органов. Автоматизация процессов минимизирует такие риски, повышая финансовую дисциплину и защищенность учреждения.

В целом, внедрение автоматизированной системы учета тарифицированных ставок преподавателей будет способствовать оптимизации управленческих процессов, снижению трудовых и финансовых издержек, а также повышению общей эффективности работы образовательного учреждения.

В основе образовательного процесса лежит преподавательская деятельность. Тарификация деятельности преподавателей позволяет определить учебную нагрузку, а учебная нагрузка в свою очередь влияет на оплату труда. Автоматизированные информационные системы позволяют эффективно управлять процессом тарификации и освободить руководителей от длительных и сложных процедур по формированию учётных данных о преподавателях и сбору информационно-справочных материалов образовательного процесса. Результаты тарификации используются для анализа образовательной деятельности учреждения и принятия управленческих решений руководством.

Целью представленной работы является разработка проекта автоматизированной системы учета тарификации преподавателей на примере ГАПОУ ТСПК Тольяттинского социального-педагогического колледжа.

Объектом исследования в данной работе является учет тарификации преподавателей.

Предметом исследования является процесс разработки проекта автоматизированной системы учета тарификации преподавателей.

Задачи представленной работы:

- выполнить анализ современных автоматизированных информационных систем (АИС) тарификации преподавателей;
- выбрать архитектуру АИС тарификации преподавателей;
- разработать прототип базы данных АИС тарификации преподавателей и проведение тестирования ее функциональности;
- выполнить проектирование пользовательского интерфейса АИС тарификации преподавателей;
- оценить экономическую эффективность использования системы тарификации преподавателей.

# Глава 1 Функциональное моделирование предметной области

## 1.1 Технико-экономическая характеристика предметной области

Название исследуемого колледжа: государственное автономное профессиональное образовательное учреждение Самарской области "Тольяттинский социально-педагогический колледж".

Сокращенное название: ГАПОУ ТСПК.

Организационная структура колледжа представлена на рисунке 1.

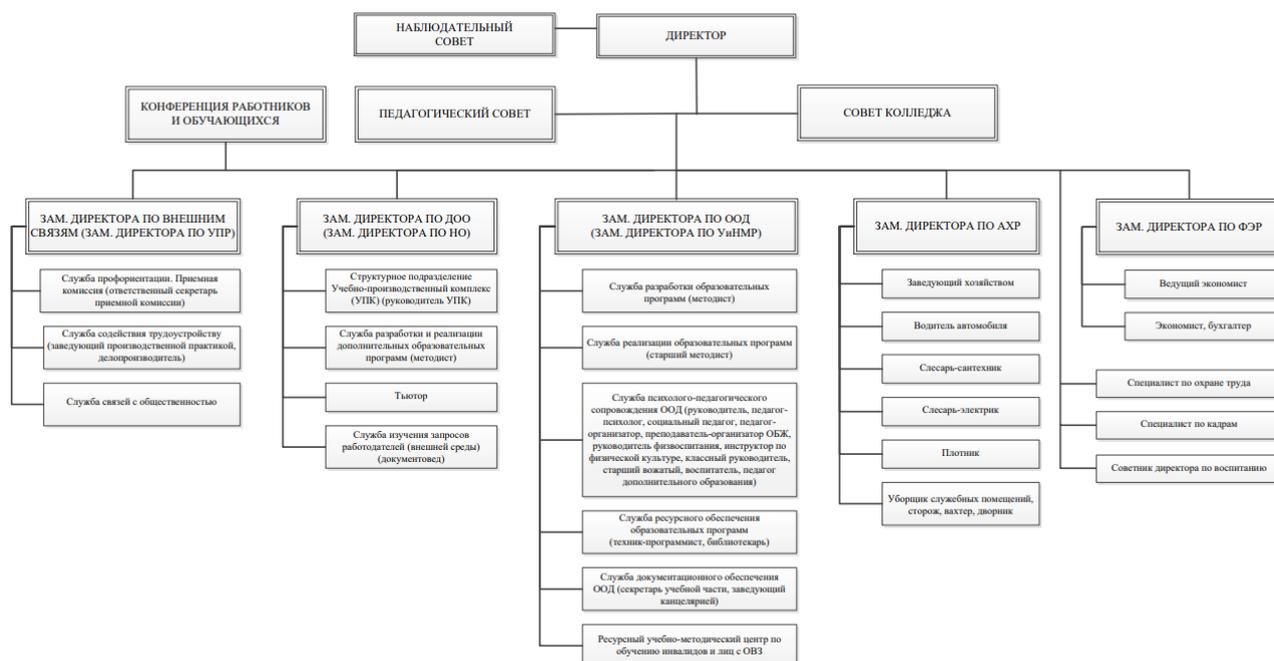


Рисунок 1 - Организационная структура колледжа

Колледж осуществляет подготовку высококвалифицированных, конкурентоспособных специалистов. Образовательный процесс в колледже осуществляется по шестидневной рабочей неделе, согласно расписанию учебных занятий. Тольяттинский социально-педагогический колледж готовит специалистов для детских садов, школ и учреждений социальной защиты, а также специалистов в сфере IT.

В колледже обучаются программисты, операторы информационных ресурсов и специалисты по информационным системам, воспитатели, учителя начальных классов, учителя физической культуры, социальные работники, педагоги в сфере дополнительного образования.

В 2020 году колледж занял первое место среди учебных заведений Самарской области по результатам эффективности деятельности.

В колледже можно получить дополнительное образование уже во время учебы по основной специальности.

Предметом исследования является процесс автоматизации тарификации преподавателей. Тарификация преподавателей — это определение учебной нагрузки и составление тарификационных списков. Учебная нагрузка влияет на оплату труда, так как на её основе рассчитывают оклады.

Заработная плата в месяц педагогического персонала, состоит из базовой части, компенсационных выплат и выплат стимулирующего характера. В состав должностного оклада учителя входят заработная плата за аудиторную занятость, доплата за обучение детей-инвалидов, а также доплата за деление класса на группы. К выплатам компенсационного характера относятся доплаты за работу в ночное время, за работу в выходные и праздничные дни, за сверхурочную работу.

Тарификация в исследуемом колледже проводится ежегодно. Весной формируется предварительный план показателей на следующий учебный год. С этими данными знакомятся преподаватели. Каждый учитель получает информацию о том, какие основные и дополнительные программы и курсы ему предстоит вести в следующем учебном году, а также о предполагаемой индивидуальной учебной нагрузке. По мере приближения начала учебного года предварительный план корректируется с учетом возможных изменений в нагрузке. Обычно к 1 октября администрация колледжа утверждает окончательный план, где указаны не только виды работ и учебные часы, но и конкретные ставки оплаты труда для каждого педагога.

При тарификации преподавателя исследуемого колледжа подсчитываются следующие показатели и особые условия:

- общая нагрузка преподавателя - количество часов, которое было потрачено на все занятия. При этом отдельно считается количество часов на занятия с бюджетными и внебюджетными группами;
- количество часов, которое преподаватель обучает студентов с ОВЗ (ограниченными возможностями здоровья);
- количество часов работы с группами углубленной подготовки;
- количество часов работы на проверку письменных работ;
- количество часов, которые преподаватель тратит на консультации, экзамены, промежуточные аттестации, ГИА, домашние контрольные работы, курсовые работы.

При расчете часовой нагрузки учитывается количество занимаемых сотрудником должностей, а также соблюдение законодательных норм относительно максимальной продолжительности рабочей недели. Важно помнить, что педагоги не могут работать более 36 часов в неделю.

Объём учебной нагрузки больше или меньше нормы устанавливается только с письменного согласия работника. При закрытии вакансии часы передаются основному работнику. Часы учебной нагрузки сверх установленной нормы оплачиваются дополнительно.

## **1.2 Концептуальное моделирование предметной области**

### **1.2.1 Выбор технологии концептуального моделирования предметной области**

Для концептуального моделирования предметной области используется инструмент – BPWin (Business Process Modeler). BPWin разработан компанией Logic Works и используется для моделирования бизнес-процессов.

Функциональные возможности BPWin:

- функциональное моделирование на основе стандарта *idef0*;
- моделирование потока работ на основе стандарта *idef3*;

- моделирование потока данных на основе стандарта dfd;
- создание смешанных моделей;
- детализация моделей для комплексного анализа работы организации;
- моделирование стоимостного анализа [3].

Для проектирования диаграмм «как есть» выберем методологию IDEF0 [5].

Описание диаграмм происходит в форме «чёрного ящика» с входами, выходами, управлением и механизмом. Стрелки входа, управления, механизма и выхода приходят в соответствующие кромки работы. Входящие стрелки показывают входные данные, которые поступают в процесс. Исходящие стрелки отражают результаты выполнения процесса. Управляющие стрелки указывают на правила, стандарты или ограничения, которые влияют на выполнение процесса [6].

Для проектирования диаграмм «как будет» выберем методологию DFD.

DFD – это диаграммы потоков данных, которые используются для визуализации и анализа информационных потоков в рамках системы или процесса [7]. DFD состоит из основных элементов:

- внешние сущности — это внешние источники и получатели данных. они представляют собой объекты, которые обмениваются информацией с системой, предоставляя ей данные или получая их. эти сущности находятся вне границ рассматриваемой диаграммы;
- процессы (обработка данных). процессы описывают функции, которые выполняют обработку входных данных и генерируют выходные данные. процессы могут быть декомпозированы на более мелкие процессы для более детального анализа;
- потоки данных (стрелки на диаграмме). потоки данных показывают направление движения информации от одного элемента к другому;
- накопители данных. они представляют собой места хранения данных, где информация может быть временно сохранена или изменена [8].

### **1.2.2 Моделирование бизнес-процессов предметной области для постановки задачи автоматизированного варианта решения**

В связи с высокой трудоемкостью обработки сведений о нагрузках преподавателей, в исследуемом колледже выявилась необходимость автоматизации процесса тарификации преподавателей. Тарификация преподавателей с использованием программного средства MS Excel не позволяет хранить информацию в удобном формате, а разрозненное хранение данных приводит к ошибкам в расчетах.

Перечислим основные бизнес-процессы исследуемой предметной области:

- учет преподавательского состава;
- учет справочных данных;
- тарификация преподавателей;
- формирование сводной ведомости.

Типовая структура бизнес-процесса включает в себя:

- исходные материалы или информация, необходимые для начала выполнения процесса;
- результаты выполнения бизнес-процесса;
- оборудование, персонал, финансы и другие средства, используемые для выполнения процесса;
- методики, инструкции, правила выполнения и требования;
- исполнители, которые задействованы в выполнении бизнес-процесса.

### **1.2.3 Разработка и анализ модели бизнес-процесса «как есть»**

Опишем существующую предметную технологию с помощью IDEF0 диаграммы. Контекстная диаграмма приведена на рисунке 2.

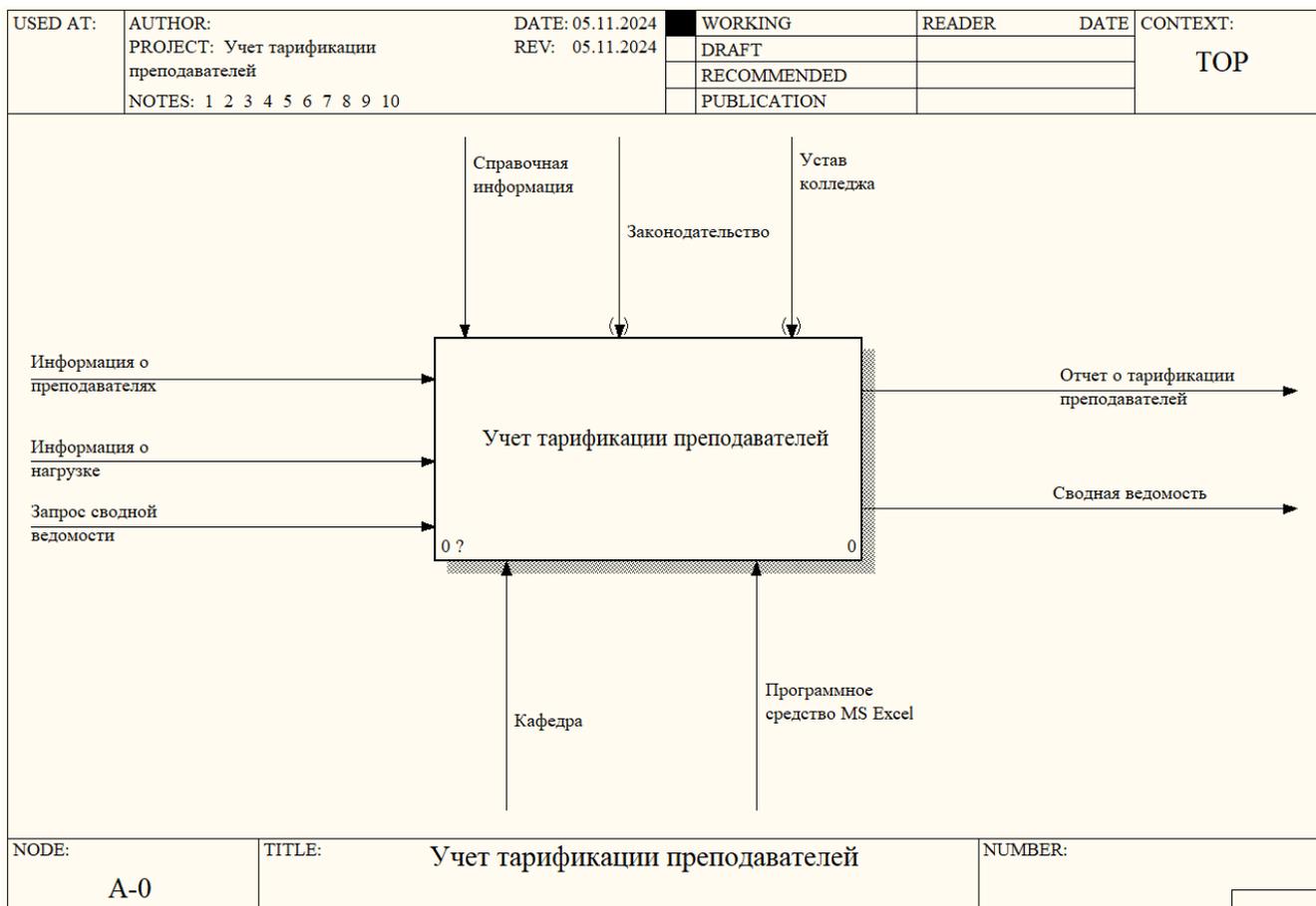


Рисунок 2 – Контекстная диаграмма

На декомпозиции первого уровня представлены следующие работы «Учет тарификации преподавателей»:

- учет преподавательского состава;
- тарификация преподавателей;
- формирование сводной ведомости.

Декомпозиция первого уровня работы «Учет тарификации преподавателей» представлена на рисунке 3.

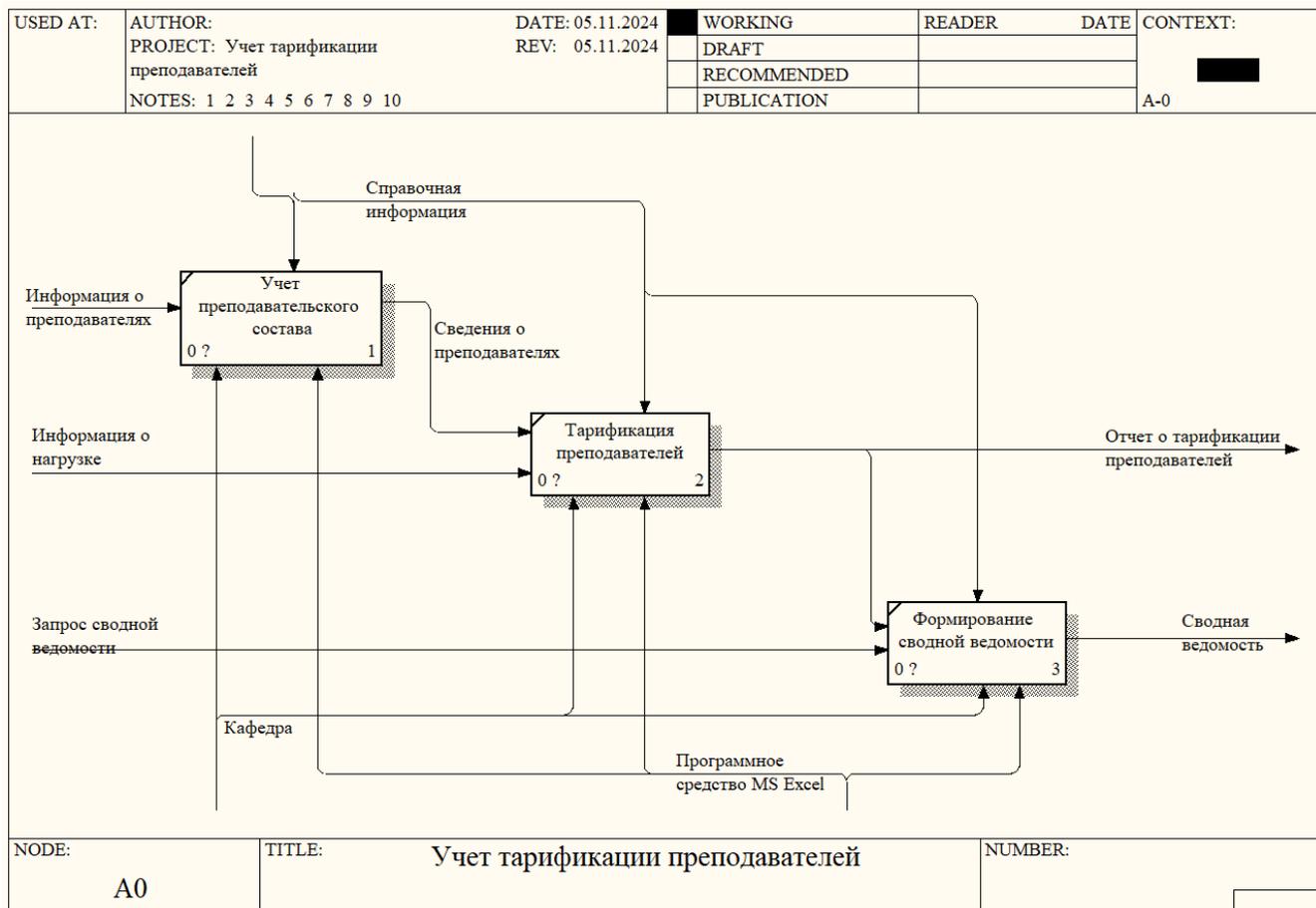


Рисунок 3 – Декомпозиция первого уровня работы «Учет тарификации преподавателей»

#### 1.2.4 Обоснование необходимости автоматизированного варианта решения и формирование требований к новой технологии

Перечислим «недостатки существующей системы обработки информации»:

- многочисленные ошибки ручного ввода данных;
- избыточность и противоречивость информации;
- затраты времени и лишних человеческих ресурсов на ручной труд;
- низкая скорость обработки информации;
- сложность обеспечения конфиденциальности и безопасности данных;
- отсутствие возможностей автоматизированного анализа и контроля данных;

– проблемы получения актуальной и полной картины преподавательской деятельности.

В рамках представленной работы необходимо разработать функционал для обеспечения учета тарификации преподавателей.

Перечислим необходимые этапы разработки web-приложения для учёта тарификации преподавателей:

– изучение требований заказчика и определение функциональных и нефункциональных требований к системе;

– разработка архитектуры и дизайна системы, проектирование структуры базы данных, макетирование интерфейса пользователя и описание логики работы приложения;

– разработка базы данных, серверной логики и клиентского интерфейса (см. Приложение А);

– тестирование web-приложения, проверка работоспособности, надёжности и соответствия требованиям системы, исправление ошибок и доработка функционала при необходимости;

– развёртывание и поддержка web-приложения, установка системы на сервере, обеспечение стабильной работы, периодическое обновление системы и предоставление поддержки пользователям;

– обучение пользователей, проведение обучающих семинаров, предоставление документации и видеоматериалов для ознакомления с функционалом и возможностями системы.

### **1.3 Анализ существующих разработок на предмет соответствия сформулированным требованиям**

#### **1.3.1 Определение критериев анализа**

В качестве критериев анализа были выбраны следующие критерии:

– настройка прав пользователей;

- удобство интерфейса пользователя;
- учет бюджетных и внебюджетных нагрузок;
- учет уровня образования;
- стоимость продукта и дальнейшее обслуживание.

### 1.3.2 Сравнительная характеристика существующих разработок

Рассмотрим программные продукты, которые представлены на рынке и подходят для решения проблем учета тарификации преподавателей. Сравнительная характеристика существующих разработок представлена в таблице 1.

Таблица 1 - Сравнительная характеристика существующих разработок

| Требования                                   | 1С:Зарплата и кадры | Модуль "Тарифицирование в образовании" — ПАРУС-Бюджет 10 | АРМ Преподаватель Цифровизация Плюс |
|--|---------------------|--|-------------------------------------|
| Настройка прав пользователей                 | +                   | +  | +                                   |
| Удобство интерфейса пользователя             | +                   | +  | -                                   |
| Учет бюджетных и внебюджетных нагрузок       | +                   | -  | +                                   |
| Учет уровня образования                      | -                   | -  | -                                   |
| Стоимость продукта и дальнейшее обслуживание | очень дорого        | дорого   | приемлемо                           |

Использование готовых решений может не всегда удовлетворять требованиям и потребностям заказчика. В результате анализа принято решение о собственной разработке информационной системы учета тарификации преподавателей.

При разработке собственной информационной системы автор может учитывать особенности и специфику работы исследуемого колледжа, а также интегрировать ее с другими имеющимися системами, что может значительно упростить рабочие процессы. Важно также учитывать технические возможности

колледжа, чтобы система была оптимизирована под имеющуюся IT-инфраструктуру.

Такой способ решения проблемы представляется автору наиболее подходящим. Это позволит разработать именно те функции, которые будут максимально полезны для решения задач и потребностей колледжа в данный момент.

#### **1.4 Постановка задачи на разработку проекта создания АИС**

Опишем функциональные требования к системе:

- работа со справочными данными (данные о группах и дисциплинах);
- работа с личными данными преподавателей колледжа;
- работа с данными по тарификации преподавателей, учет нагрузки преподавателей, учет особых условий. фильтр и экспорт данных по тарификации в excel;
- формирование сводной ведомости по преподавателям колледжа. фильтр и экспорт данных сводной ведомости в excel.

#### **1.5 Разработка модели бизнес-процесса «как должно быть»**

Модель бизнес-процессов «как должно быть» построена в нотации DFD.

На модели располагается внешняя сущность «Пользователь» и главная работа «Учет тарификации преподавателей».

В качестве входной информация поступают следующие потоки:

- информация о преподавателях;
- информация о нагрузке;
- запрос сводной ведомости.

В качестве выходной информация используются следующие потоки:

- отчет о тарификации преподавателей;
- сводная ведомость.

Контекстная диаграмма приведена на рисунке 4.

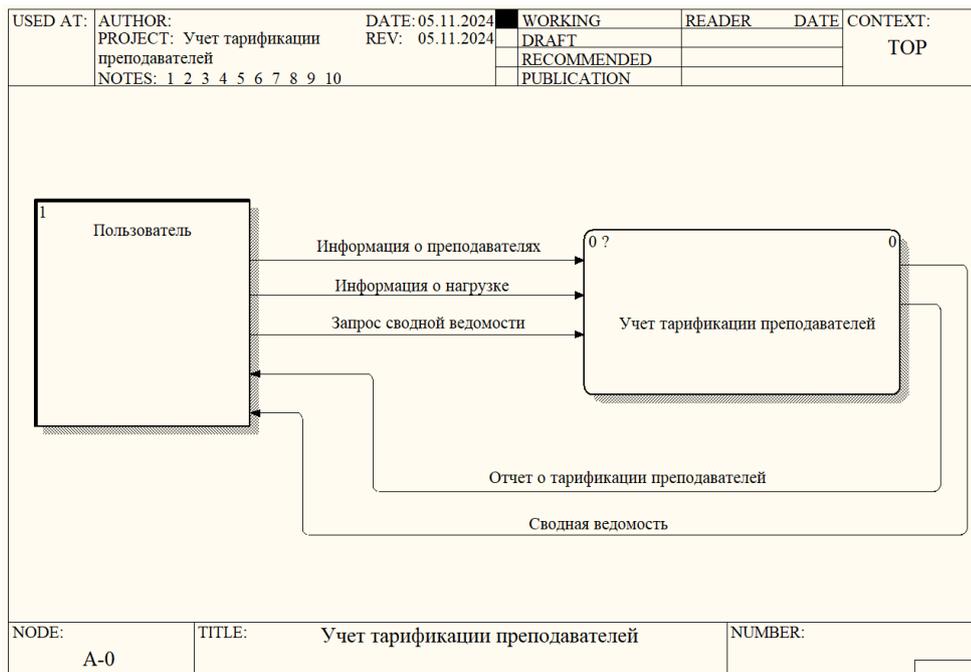


Рисунок 4 - Контекстная диаграмма процесса ТО-ВЕ

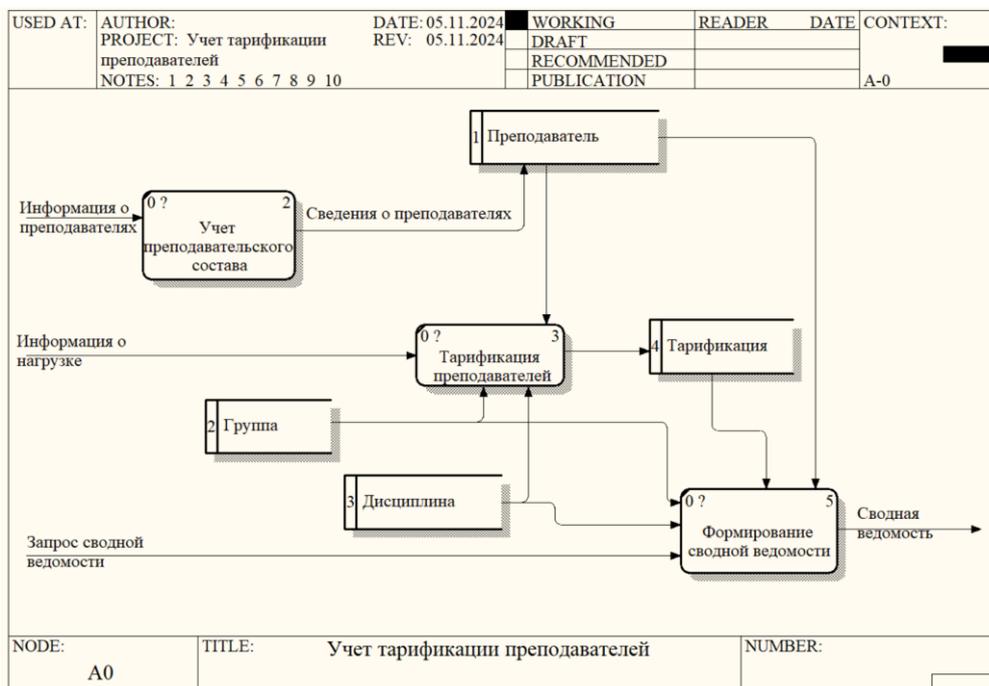


Рисунок 5 - Декомпозиция процесса «Учет тарификации преподавателей»

На модели декомпозиции используются следующие хранилища:

- преподаватель;
- группа;
- дисциплина;
- тарификация.

Разрабатываемое web-приложение должно позволить снизить влияние человеческого фактора на ведение документации по учету информации о тарификации и минимизировать временные затраты на операции, связанные с учетом, обработкой и предоставлением необходимых данных.

Перечислим преимущества обработки информации с помощью автоматизированной системы:

- повышение эффективности работы (меньше ручного труда);
- уменьшения нагрузки на преподавателей, вместо рутинной работы по оформлению отчетов преподаватели будут иметь возможность уделить больше внимания непосредственно образовательной деятельности;
- увеличение производительности (высокая скорость работы после автоматизации);
- повышения точности и своевременности выполнения работ по тарификации преподавателей;
- снижение затрат (после внедрения понадобится меньше сотрудников);
- возможности анализа (предоставление руководству инструментов для быстрого получения информации о процессе тарификации преподавателей);
- улучшения качества контроля и проверки результатов тарификации;
- экономия бюджета колледжа, отсутствие оплаты за сопровождение программного продукта;
- уникальность (возможность создать конкурентоспособное ИТ-решение, учитывающее особенности исследуемого колледжа);
- окупаемость вложений (возможность окупить инвестиции в разработку и получать прибыль в будущем).

## Выводы по главе 1

В первой главе представлена технико-экономическая характеристика исследуемого колледжа, которая включает в себя анализ текущего состояния образовательного учреждения с точки зрения его экономической эффективности и технической оснащенности. В рамках этого анализа была построена организационная структура колледжа, а также разработаны две ключевые модели:

Модель «КАК ЕСТЬ» – отражает текущее состояние системы управления колледжем, включая существующие процессы, механизмы взаимодействия между подразделениями и уровень автоматизации различных операций.

Модель «КАК ДОЛЖНО БЫТЬ» – представляет собой целевую модель, описывающую идеальное состояние системы после внедрения изменений и улучшений. Эта модель демонстрирует, каким образом колледж может оптимизировать свои процессы, повысить эффективность работы и улучшить качество предоставляемых услуг.

Кроме того, были проанализированы уже существующие на рынке готовые решения, подходящие для автоматизации процессов учета тарификации преподавателей. Эти решения оценивались с точки зрения их функциональных возможностей, соответствия требованиям колледжа и потенциальной выгоды от их использования.

На основе проведенного анализа были поставлены конкретные задачи на разработку новой автоматизированной информационной системы (АИС) тарификации преподавателей. Эта система должна будет обеспечить более точный учет рабочего времени преподавателей, автоматизацию расчета заработной платы и повышение прозрачности процесса тарификации.

## Глава 2 Логическое проектирование АИС

### 2.1 Выбор технологии логического моделирования АИС

В качестве логической модели АИС было принято построить диаграмму вариантов использования.

Диаграмма вариантов использования является одной из основных UML-диаграмм. Перечислим назначения диаграммы вариантов использования:

- описание функциональности системы: диаграмма служит для описания функций, которые могут быть выполняются в системе.
- понимание потребностей и требований пользователей: диаграмма может помочь проектировщикам лучше понять потребности и требования пользователей к системе.
- определение потоков данных: диаграмма вариантов использования может использоваться для определения потоков данных между вариантами использования и элементами системы.
- распределение задач в команде: диаграмма может помочь членам команды лучше понять, какие задачи они должны выполнять, и как они должны взаимодействовать между собой.
- тестирование системы: диаграмма служит для определения тестовых случаев и тестовых сценариев системы [4].

Построим диаграмму вариантов использования с помощью программного средства MS Visio. MS Visio - это программное средство для создания диаграмм и графических схем. Оно позволяет пользователям создавать различные типы диаграмм, такие как блок-схемы, графики, карты и схемы баз данных.

Основные возможности MS Visio включают в себя:

- широкий набор инструментов для рисования и редактирования диаграмм и схем.
- встроенные функции для проверки правописания, грамматики и других ошибок.

— возможность экспорта диаграмм в различные форматы, такие как pdf, png и другие [6].

## 2.2 Логическая модель АИС и её описание

Логическая модель [20] АИС представлена на рисунке 6.



Рисунок 6 - Логическая модель АИС

Перечислим варианты использования модели:

- авторизация пользователя.
- работа со справочниками (группы и дисциплины).
- работа с преподавателями.
- работа с тарификацией.
- формирование отчетных форм (сводная ведомость и тарификация преподавателей).

## 2.3 Информационное обеспечение АИС

### 2.3.1 Используемые классификаторы и системы кодирования

Состав классификаторов представлен в таблице 2.

Таблица 2 – Состав классификаторов

| Наименование        | Длина кода | Система кодирования | Вид           | Структура |
|---------------------|------------|---------------------|---------------|-----------|
| Номер преподавателя | 4          | Порядковая          | Общесистемный | XXXXXX    |
| Номер группы        | 4          | Порядковая          | Общесистемный | XXXXXX    |
| Номер дисциплины    | 4          | Порядковая          | Общесистемный | XXXXXX    |

### **2.3.2 Характеристика нормативно-справочной и входной оперативной информации**

В качестве входной информации для разрабатываемого приложения поступают следующие данные:

- данные о преподавателе.
- данные о учебных группах.
- данные о дисциплинах коллежа.
- данные о тарификации преподавателей.

### **2.3.3 Характеристика выходной информации**

В качестве выходной информации формируются следующие данные:

- экспорт данных о тарификации в excel.
- экспорт сводной ведомости в excel.

## **2.4 Проектирование базы данных АИС**

### **2.4.1 Выбор технологии проектирования БД АИС**

Инфологическое проектирование выполнено с помощью инструмента для проектирования и документирования баз данных ERWin.

Перечислим возможности ERWin [7], [18]:

- моделирование данных с использованием стандартов IDEF1X и IE.

- анализ моделей, поиск ошибок и несоответствий.
- интеграция с другими инструментами и технологиями, такими как Microsoft Visio, IBM Rational Rose и Oracle Designer.
- генерация кода на языке SQL.
- поддержка различных платформ и СУБД, включая Microsoft SQL Server, Oracle Database, MySQL, Access [1].
- в качестве методологии проектирование базы данных была выбрана IDEF1X [10], [11].

#### 2.4.2 Разработка концептуальной модели данных АИС

Концептуальная модель данных представлена на рисунке 7.

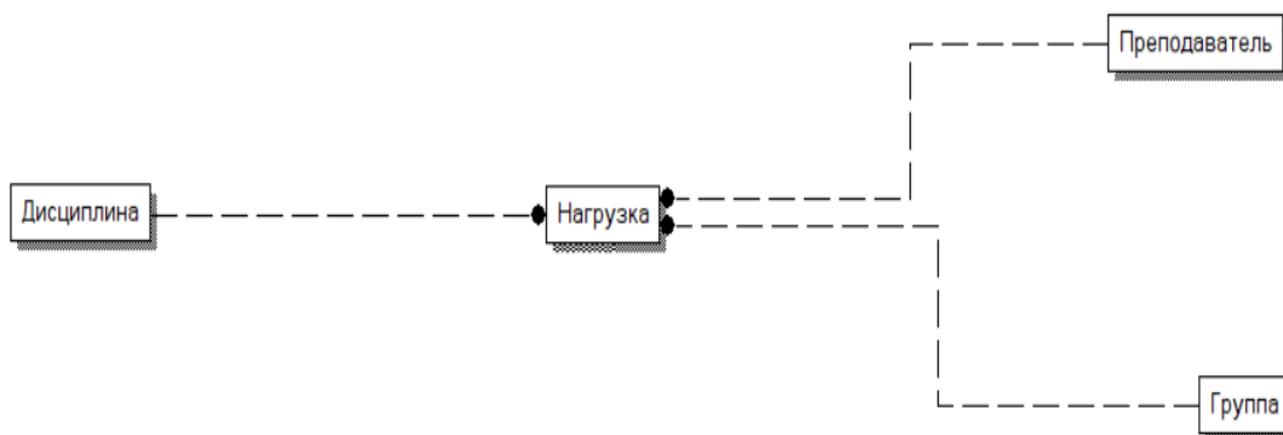


Рисунок 7 - Концептуальная модель данных

#### 2.4.3 Обоснование вида логической модели

«В качестве модели данных была выбрана реляционная модель базы данных. Перечислим преимущества реляционной модели базы данных» [17]:

- модель позволяет создавать таблицы с различными связями между данными, что позволяет гибко управлять информацией;
- модель проста в использовании и понимании, что делает ее доступной для любого пользователя;

- модель обеспечивает быстрый доступ к данным, что позволяет повысить производительность базы данных;
- модель обеспечивает целостность данных, что позволяет избежать ошибок и повреждения информации;
- модель обеспечивает управление доступом к данным, что позволяет защитить информацию от несанкционированного доступа;
- модель имеет широкую поддержку со стороны различных программных продуктов, что облегчает ее использование и разработку [12].

#### 2.4.4 Разработка логической модели данных АИС

Логическая модель данных представлена на рисунке 8.

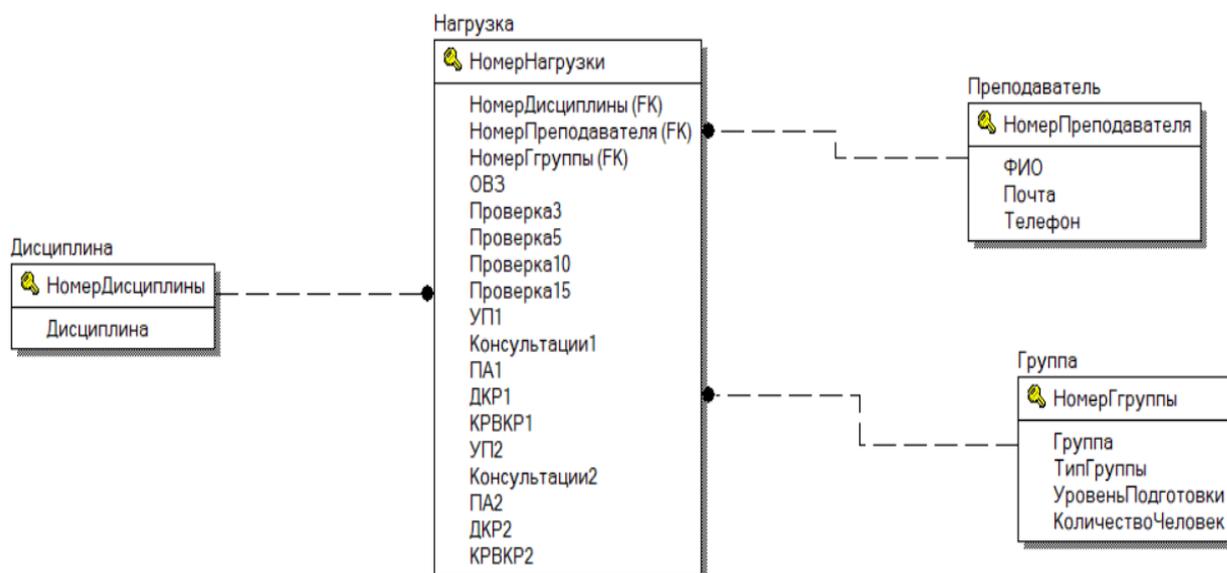


Рисунок 8 - Логическая модель данных

#### 2.5 Требования к аппаратно-программному обеспечению АИС

Теперь перечислим технические характеристики оборудования для сервера:

- процессор intel core i5 4460;
- ОЗУ – 16 Гб;
- свободное место на жестком диске – 80 Гб.

Технические характеристики оборудования для рабочей станции:

- процессор intel core i3 2236;
- ОЗУ – 8 Гб;
- свободное место на жестком диске – 50 Гб.

Программное обеспечение для сервера включает:

- ОС Linux;
- сервер Apache и MySQL;
- браузер Mozilla Firefox.

Перечислим конфигурации программных средств для рабочей станции (клиента):

- MS Windows 7;
- MS Office;
- Mozilla Firefox.

Выводы по главе 2

Во второй главе представлен результат опытно-практической работы, направленной на создание информационной модели и проектирование базы данных для разрабатываемой автоматизированной информационной системы (АИС).

Информационная модель была разработана с учетом всех требований, выявленных при анализе предметной области. Описание информационной модели включает в себя детальный разбор каждого компонента системы, его назначение и роль в общем процессе.

Проектирование базы данных является ключевым этапом разработки любой информационной системы. В данной работе подробно описан процесс создания структуры базы данных, включающий определение таблиц, полей, связей между ними, а также выбор подходящих типов данных для хранения информации. Особое внимание уделено вопросам обеспечения целостности данных и безопасности доступа к ним.

Также в этой главе перечислены требования к аппаратно-программному обеспечению, необходимому для успешной реализации разработанной системы.

Указаны минимальные и рекомендуемые характеристики оборудования, такие как мощность процессора, объем оперативной памяти, емкость жесткого диска и другие параметры, обеспечивающие стабильную работу системы. Кроме того, описаны требования к программному обеспечению: операционные системы, СУБД, средства разработки и тестирования, а также дополнительные утилиты, которые могут потребоваться для эксплуатации и поддержки системы.

Таким образом, вторая глава закладывает основу для практической реализации проекта, обеспечивая четкое понимание структуры данных и технических условий, необходимых для успешного внедрения автоматизированной системы тарификации преподавателей.

## Глава 3 Физическое проектирование АИС

### 3.1 Выбор архитектуры АИС

В качестве архитектуры АИС была выбрана архитектура «клиент-сервер», перечислим её преимущества:

- централизованное управление - надёжный контроль над данными и ресурсами упрощает администрирование и обновление информации;
- распределение нагрузки - одновременная обработка нескольких запросов от разных клиентов позволяет рационально использовать ресурсы;
- безопасность - доступ к данным контролируется центральной системой, обеспечивая авторизацию и аутентификацию пользователей. Разделение клиентских и серверных компонентов позволяет лучше контролировать доступ к данным и приложениям, что повышает уровень безопасности системы;
- лёгкое масштабирование и обновление системы - система легко адаптируется к увеличению количества клиентов и данных;
- высокая надёжность - централизованное хранение данных на сервере гарантирует их защиту;
- разделение функций - клиенты отвечают за представление данных и пользовательский интерфейс, а серверы обрабатывают бизнес-логику и хранение данных. Это упрощает разработку и поддержку системы;
- повышение производительности: разделение функций между клиентами и серверами позволяет оптимизировать обработку данных и уменьшить нагрузку на серверы [13].

Архитектура системы представлена на рисунке 9.

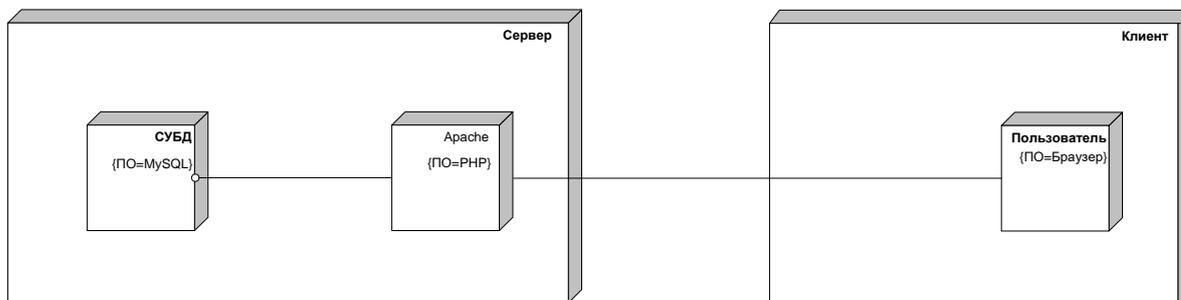


Рисунок 9 – Архитектура системы

Серверная часть состоит из:

1. Сервер базы данных с установленной MySQL;
2. Web-сервер со следующей конфигурацией ПО:
  - 2.1 Операционная система – Windows 7/10;
  - 2.2 Http-сервер Apache;
  - 2.3 Транслятор PHP.

Клиентская часть системы представлена браузером пользователя.

### 3.2 Выбор технологии разработки программного обеспечения АИС

Для начала выберем язык программирования.

Приведём в таблице 3, сравнительную оценку языков программирования.

Таблица 3 - Сравнительная таблица языков программирования

|    | Критерий                         | C#      | Perl    | Java    | PHP     |
|----|----------------------------------|---------|---------|---------|---------|
| 1. | Уровень сложности изучения языка | Низкий  | Низкий  | Высокий | Низкий  |
| 2. | Распространённость               | Низкая  | Низкая  | Низкая  | Высокая |
| 3. | Компактность кода                | Высокая | Высокая | Низкая  | Высокая |
| 4. | Кроссплатформенность             | Нет     | Нет     | Да      | Да      |
| 5. | Функциональность                 | Высокая | Высокая | Высокая | Высокая |
| 6. | Удобство применения              | Высокая | Высокая | Низкая  | Высокая |

Перечислим основные преимущества языка программирования PHP:

- простота: PHP легко изучить, и для освоения основ достаточно 1–1,5 месяца.
- гибкость: подходит для разных задач, включая небольшие и крупные проекты.
- лёгкий доступ: бесплатный доступ и возможность использования любого редактора кода.
- живое сообщество: множество гайдов, форумов и помощи от опытных коллег.
- универсальность: подходит для всех веб-серверов.
- лёгкая интеграция с базами данных, такими как MySQL, PostgreSQL и SQLite.
- поддержка множества фреймворков и CMS, таких как Laravel, Symfony, WordPress и Joomla [14].

### 3.3 Выбор СУБД АИС

Для выбора были рассмотрены три свободно распространяемые СУБД: Firebird, MySQL и PostgreSQL [15], [19].

Сравнение СУБД представлено в таблице 4.

Таблица 4 - Сравнение параметров СУБД

| Критерий/СУБД           | MySQL                          | PostgreSQL     | FirebirdSQL    |
|-------------------------|--------------------------------|----------------|----------------|
| Поддерживаемые ОС       | Windows, Linux                 | Windows, Linux | Windows, Linux |
| Среды разработки        | phpMyAdmin,<br>MySQL Workbench | phpPgAdmin     | Ibexpert       |
| Триггеры и ХП           | Есть                           | Есть           | Есть           |
| Средства резервирования | Есть                           | Есть           | Есть           |
| Средства репликации     | Есть                           | Есть           | Нет            |
| Производительность      | Высокая                        | Средняя        | Средняя        |

В результате анализа в качестве СУБД была выбрана MySQL.

Перечислим основные преимущества данной СУБД:

- высокая скорость работы;
- поддержка CMS;
- бесплатная лицензия;
- простая и надёжная система безопасности;
- поддержка нескольких типов таблиц: MyISAM, InnoDB;
- плагины для упрощения и настройки работы;
- универсальность: применяется при разработке веб-сайтов, веб-приложений и корпоративных баз данных начального уровня;
- поддержка и активное обновление СУБД [9].

### 3.4 Разработка физической модели данных АИС

Для развертывания локального web-сервера и сервера базы данных MySQL устанавливаем программное средство openserver. Настройки openserver представлены на рисунке 10.

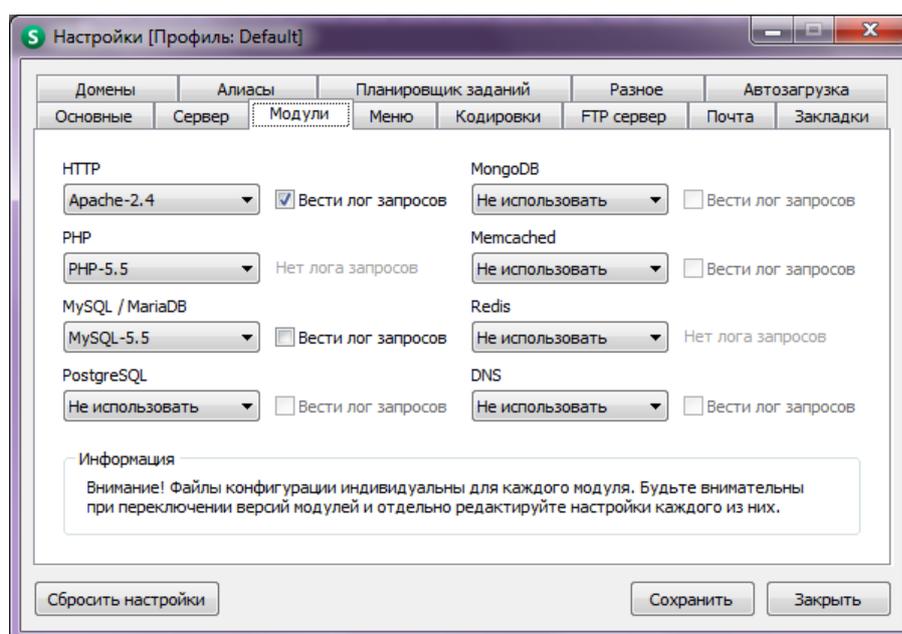


Рисунок 10 – Настройки openserver

Создаем с помощью phpmyadmin новую базу данных tariffication, рисунок 11.

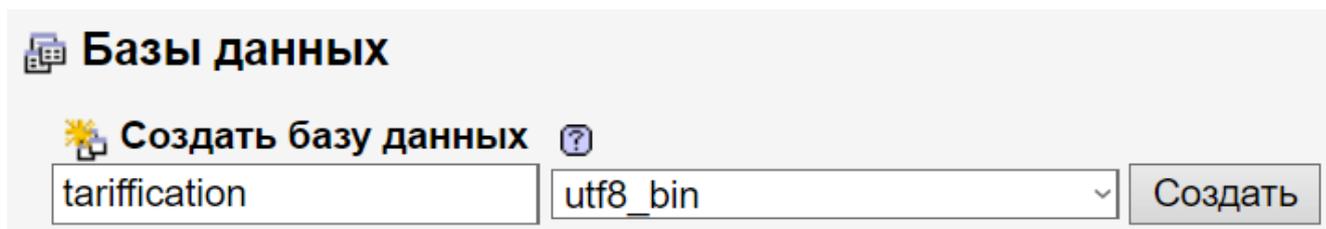


Рисунок 11 – Создание базы данных

Далее импортируем базу из файла tariffication.sql, рисунок 12.

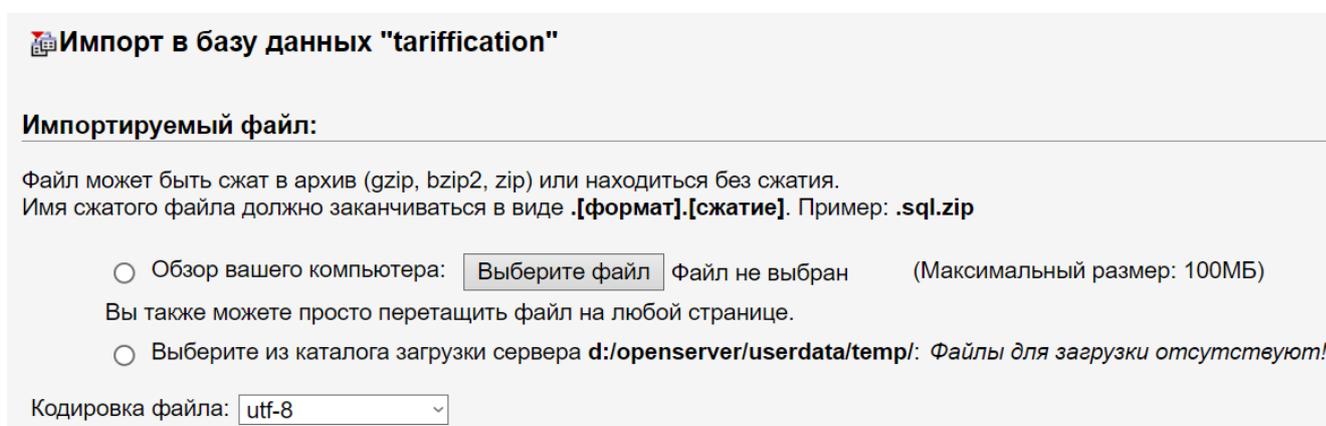


Рисунок 12 – Импорт базы данных

На рисунке 13 представлены таблицы базы данных tariffication.

|                          | Таблица ▲     | Действие   | Строки ? | Тип    | Сравнение | Размер  | Фрагментировано |
|--------------------------|---------------|------------|----------|--------|-----------|---------|-----------------|
| <input type="checkbox"/> | squad         | ☆ [иконки] | 0        | InnoDB | utf8_bin  | 16 КиБ  | -               |
| <input type="checkbox"/> | subject       | ☆ [иконки] | 0        | InnoDB | utf8_bin  | 16 КиБ  | -               |
| <input type="checkbox"/> | tariffication | ☆ [иконки] | 0        | InnoDB | utf8_bin  | 64 КиБ  | -               |
| <input type="checkbox"/> | teacher       | ☆ [иконки] | 0        | InnoDB | utf8_bin  | 16 КиБ  | -               |
|                          | 4 таблицы     | Всего      | 0        | InnoDB | utf8_bin  | 112 КиБ | 0 Байт          |

Рисунок 13 – Таблицы базы данных tariffication

На рисунке 14 представлена схема базы данных tariffication.

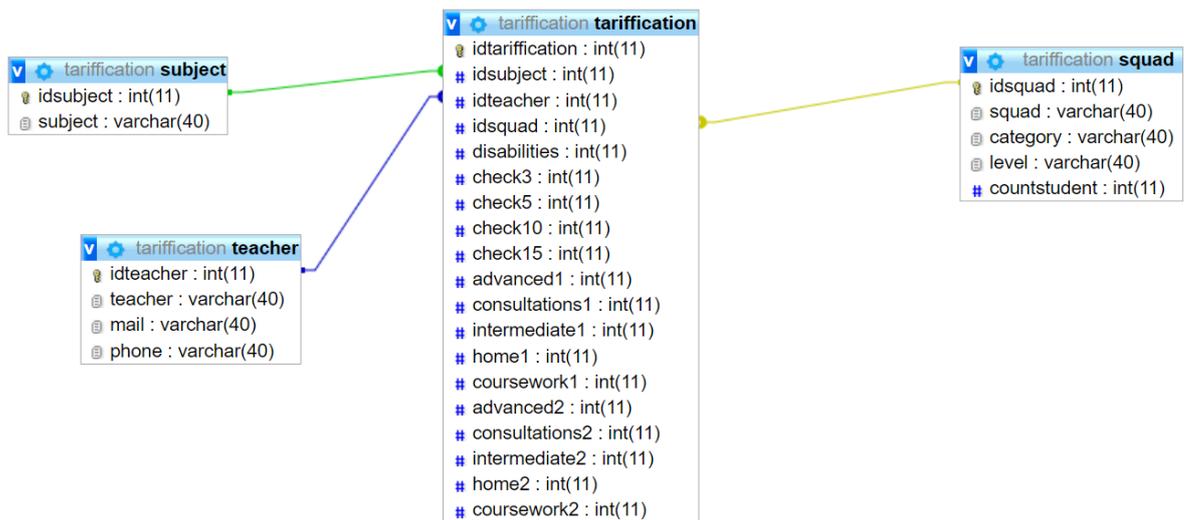


Рисунок 14 – Схема базы данных tariffication

### 3.5 Разработка программного обеспечения АИС

#### 3.5.1 Схема взаимосвязи модулей приложения АИС

Для иллюстрации взаимосвязи модулей приложения построим диаграмму вариантов использования.

Перечислим назначение диаграммы вариантов использования:

- понимание потребностей и требований пользователей: диаграмма может помочь проектировщикам лучше понять потребности, функций и требования пользователей к системе.
- определение вариантов использования: диаграмма позволяет определить различные варианты использования системы в зависимости от потребностей пользователей.
- архитектурное проектирование: диаграмма служит для создания общей архитектуры системы и определения функциональных блоков, включенных в систему.
- определение потоков данных: диаграмма вариантов использования может использоваться для определения потоков данных между вариантами использования и элементами системы.

- распределение задач в команде: диаграмма может помочь членам команды лучше понять, какие задачи они должны выполнять, и как они должны взаимодействовать между собой.
- тестирование системы: диаграмма служит для определения тестовых случаев и тестовых сценариев системы [8].

Диаграмма компонентов представлена на рисунке 15.

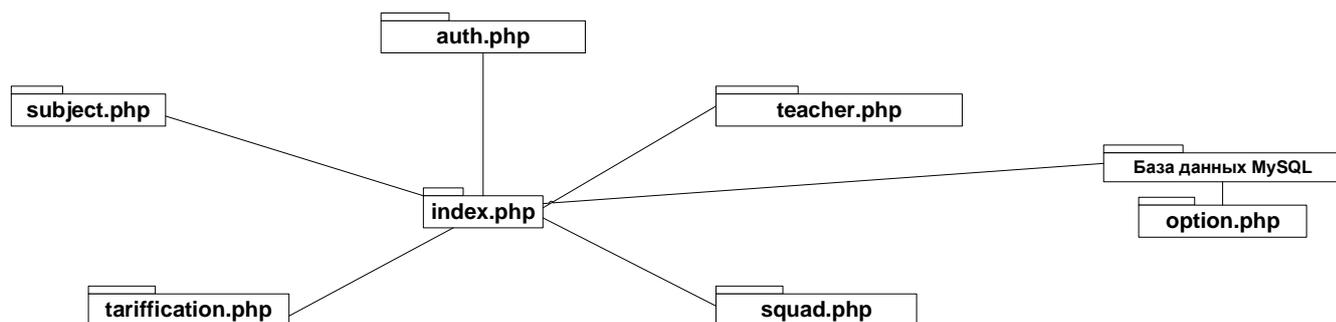


Рисунок 15 – Диаграмма компонентов

Код программы приведен в приложении А.

### 3.5.2 Описание модулей приложения АИС с примерами программного кода

Описание модулей представлено в таблице 5.

Таблица 5 – Описание модулей

| Модуль            | Описание                                  |
|-------------------|---|
| subject.php       | Работа с дисциплинами                     |
| squad.php         | Работа с учебными группами                |
| teacher.php       | Работа с преподавателями                  |
| tariffication.php | Работа с тарификацией и сводной нагрузкой |
| option.php        | Подключение к БД                          |
| index.php         | Работа главной формы                      |
| auth.php          | Авторизация пользователя                  |

### 3.6 Описание функциональности АИС

По запуску АИС на экран будет выведена стартовая страница web-приложения. Страница представлена на рисунке 16.

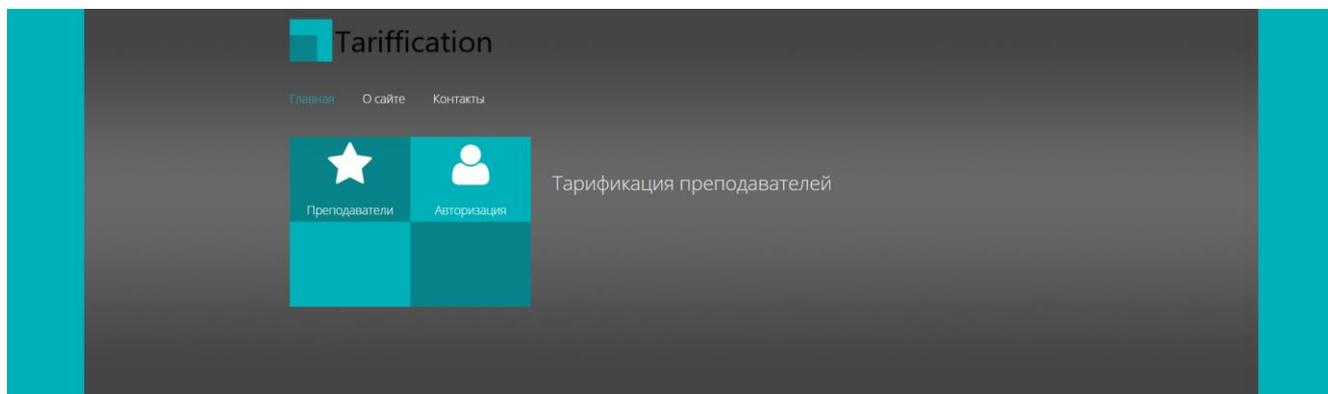


Рисунок 16 – Стартовая страница приложения

Новый пользователь, еще не прошедший процедуру авторизации, сможет ознакомиться с информацией о сайте (рисунок 17), функциональными возможностями веб-приложения (рисунок 18) и контактными данными колледжа (рисунок 19).

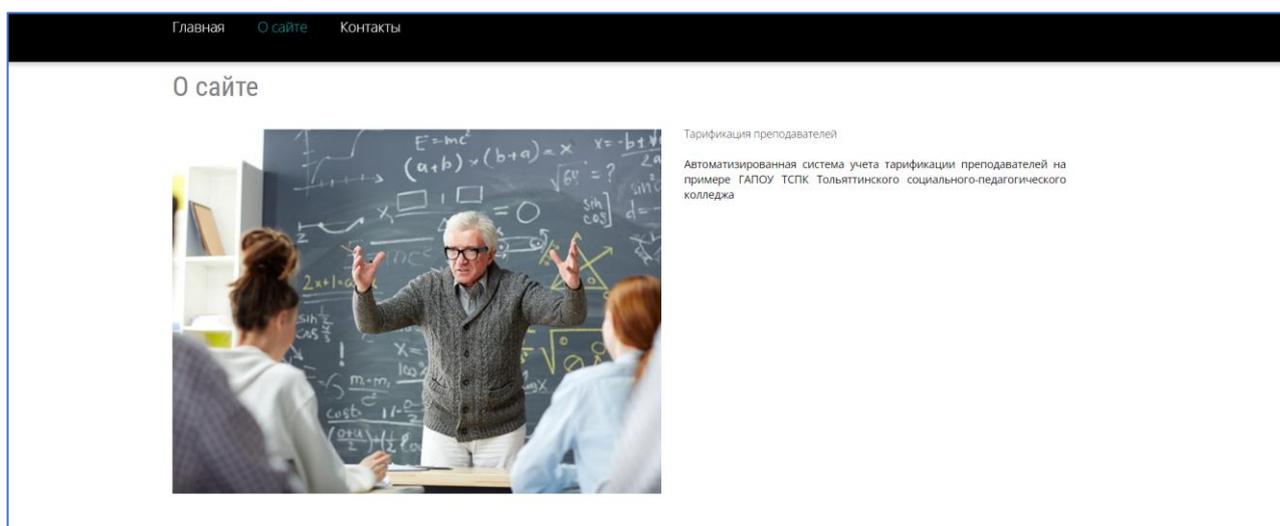


Рисунок 17 – Информация о сайте

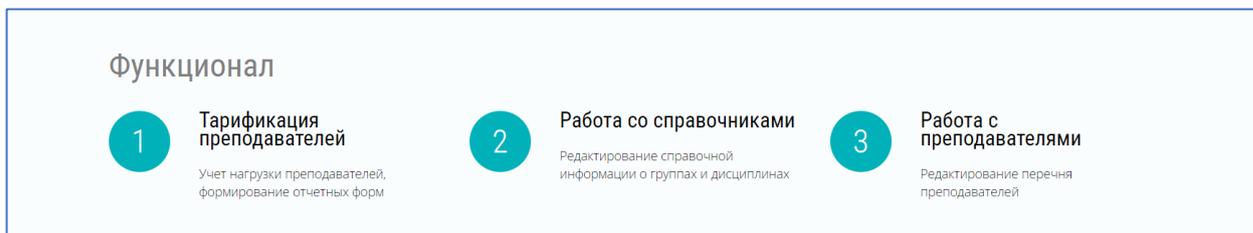


Рисунок 18 – Информация о функционале web-приложения



Рисунок 19 – Страница с контактной информацией

Форма авторизации представлена на рисунке 20.

**Авторизация**

Авторизация:  
Логин:   
Пароль:

Рисунок 20 – Форма авторизации

Ошибка авторизации представлена на рисунке 21.

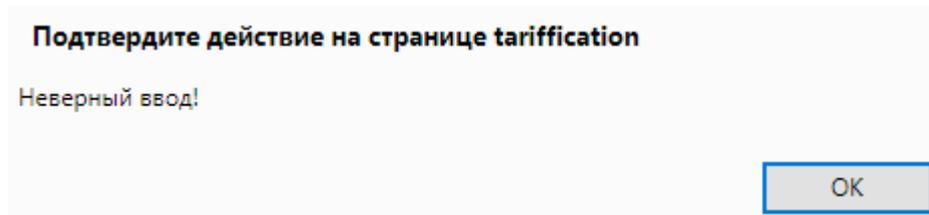


Рисунок 21 – Ошибка авторизации

В случае успешной авторизации пользователю выдаются права администратора. Авторизованному администратору доступно главное меню web-приложения. Главное меню представлено на рисунке 22.

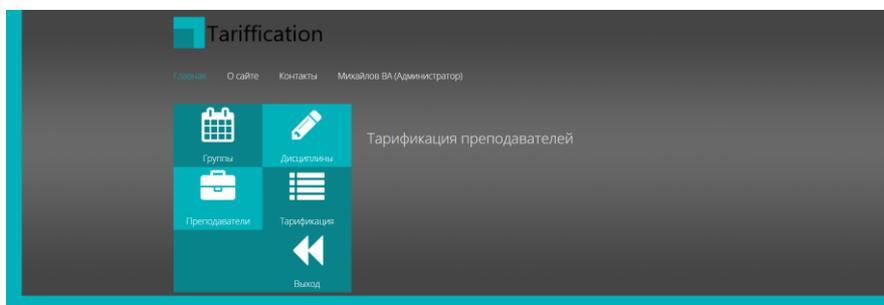


Рисунок 22 – Главное меню web-приложения

Главное меню web-приложения включает в себя следующие пункты:

1. «Группы» - работа с данными о группах, в которых обучаются студенты колледжа.
2. «Дисциплины» - работа с данными о предметах, которые преподаются в колледже.
3. «Преподаватели» - работа с данными о преподавателях, которые преподают в колледже. Именно на этот перечень преподавателей и осуществляется расчет нагрузки, выполняется процедура тарификации и составляется сводная ведомостью
4. «Тарификация» - работа с данными о тарификации преподавателей и формирование сводной нагрузки преподавателей.

Страницы для работы со справочными данными (группы и дисциплины) представлены на рисунках 23-26.

### Группы (режим администратора)

Сортировка:  Группа:

| <input type="button" value="Добавить"/> | <input type="button" value="Редактировать"/> | <input type="button" value="Удалить"/> | Группа  | Тип группы     | Уровень подготовки |
|---|--|--|---------|----------------|--------------------|
| <input checked="" type="radio"/>        |  |  | АФК-12  | Коммерческая   | Углубленная        |
| <input type="radio"/>                   |  |  | Д-11    | Коммерческая   | Базовая            |
| <input type="radio"/>                   |  |  | Д-12    | Некоммерческая | Углубленная        |
| <input type="radio"/>                   |  |  | Д-14    | Коммерческая   | Базовая            |
| <input type="radio"/>                   |  |  | ДО-51В  | Коммерческая   | Базовая            |
| <input type="radio"/>                   |  |  | ИСИП-22 | Некоммерческая | Углубленная        |
| <input type="radio"/>                   |  |  | НК-25А  | Коммерческая   | Углубленная        |

Рисунок 23 – Работа с группами колледжа

### Добавление группы (режим администратора)

Группа:

Тип группы:

Уровень подготовки:

Рисунок 24 – Добавление группы

### Дисциплины (режим администратора)

Сортировка:  Дисциплина:

| <input type="button" value="Добавить"/> | <input type="button" value="Редактировать"/> | <input type="button" value="Удалить"/> | Дисциплина             |
|---|--|--|------------------------|
| <input checked="" type="radio"/>        |  |  | Естествознание: физика |
| <input type="radio"/>                   |  |  | Литература             |
| <input type="radio"/>                   |  |  | ОБЖ                    |
| <input type="radio"/>                   |  |  | Родная литература      |
| <input type="radio"/>                   |  |  | Русский язык           |
| <input type="radio"/>                   |  |  | Физическая культура    |

Рисунок 25 – Работа с дисциплинами колледжа

## Добавление дисциплины (режим администратора)

Дисциплина:

Рисунок 26 – Добавление дисциплины

Работа с перечнем преподавателей представлена на рисунках 27-28.

## Преподаватели (режим администратора)

Сортировка:  ФИО:

|                                  | ФИО             | Телефон | Почта     |
|----------------------------------|-----------------|---------|-----------|
| <input checked="" type="radio"/> | Абасов Р.Г.     | 456456  | qwe@ya.ru |
| <input type="radio"/>            | Абрамова О.А.   | -       | asd@ya.ru |
| <input type="radio"/>            | Антропова Н.Л.  |         |           |
| <input type="radio"/>            | Божедомова С.В. |         |           |
| <input type="radio"/>            | Маркушина О.А.  |         |           |
| <input type="radio"/>            | Мастерова Л.З.  |         |           |

Рисунок 27 – Работа с перечнем преподавателей

## Добавление преподавателя (режим администратора)

ФИО:

Телефон:

Почта:

Рисунок 28 – Добавление преподавателя

Работа с тарификацией преподавателей представлена на рисунке 29.





Просмотр сводной нагрузки представлен на рисунке 33.

Сводная нагрузка (режим администратора)

Сортировка: ФИО | ФИО: Все | Фильтр | Очистить

| Преподаватель   | Нагрузка |        |           | Доплаты (от количества часов) |        |                   |    |     |     |           |        |                   |    |     |     |    |   |   |
|-----------------|----------|--------|-----------|-------------------------------|--------|-------------------|----|-----|-----|-----------|--------|-------------------|----|-----|-----|----|---|---|
|                 | Всего    | Бюджет | Внебюджет | Бюджет                        |        |                   |    |     |     | Внебюджет |        |                   |    |     |     |    |   |   |
|                 |          |        |           | ОВЗ                           | Углубл | Проверка тетрадей |    |     |     | ОВЗ       | Углубл | Проверка тетрадей |    |     |     |    |   |   |
|                 |          |        |           |                               |        | 3%                | 5% | 10% | 15% |           |        | 3%                | 5% | 10% | 15% |    |   |   |
| Абасов Р.Г.     | 61       | 61     | 0         | 61                            | 0      | 0                 | 20 | 0   | 0   | 0         | 0      | 0                 | 0  | 0   | 0   | 0  | 0 | 0 |
| Абрамова О.А.   | 202      | 151    | 51        | 0                             | 0      | 0                 | 0  | 20  | 0   | 0         | 0      | 51                | 0  | 0   | 40  | 0  |   |   |
| Антропова Н.Л.  | 71       | 0      | 71        | 0                             | 0      | 0                 | 0  | 0   | 0   | 0         | 0      | 71                | 0  | 0   | 0   | 50 |   |   |
| Божедомова С.В. | 55       | 55     | 0         | 0                             | 55     | 0                 | 0  | 0   | 12  | 0         | 0      | 0                 | 0  | 0   | 0   | 0  |   |   |
| Маркушина О.А.  | 42       | 42     | 0         | 0                             | 42     | 0                 | 22 | 0   | 0   | 0         | 0      | 0                 | 0  | 0   | 0   | 0  |   |   |
| Мастерова Л.З.  | 0        | 0      | 0         | 0                             | 0      | 0                 | 0  | 0   | 0   | 0         | 0      | 0                 | 0  | 0   | 0   | 0  |   |   |

Рисунок 33 – Просмотр сводной нагрузки

Экспорт данных о сводной нагрузке представлен ан рисунке 34.

Сводная нагрузка (1).xls - Excel

Файл Главная Вставка Разметка страницы Формулы Данные Рецензирование Вид Настройка Справка Ассистент Команда Помощь Поделиться

Буфер обмена | Шрифт | Выравнивание | Число | Стили | Ячейки | Редактирование

| Сводная нагрузка |          |        |           |                               |        |          |    |     |     |           |        |          |    |     |     |    |
|------------------|----------|--------|-----------|-------------------------------|--------|----------|----|-----|-----|-----------|--------|----------|----|-----|-----|----|
| Преподаватель    | Нагрузка |        |           | Доплаты (от количества часов) |        |          |    |     |     |           |        |          |    |     |     |    |
|                  | Всего    | Бюджет | Внебюджет | Бюджет                        |        |          |    |     |     | Внебюджет |        |          |    |     |     |    |
|                  |          |        |           | ОВЗ                           | Углубл | Проверка |    |     |     | ОВЗ       | Углубл | Проверка |    |     |     |    |
|                  |          |        |           |                               |        | 3%       | 5% | 10% | 15% |           |        | 3%       | 5% | 10% | 15% |    |
| Абасов Р.Г.      | 170      | 170    | 0         | 61                            | 0      | 0        | 20 | 0   | 0   | 0         | 0      | 0        | 0  | 0   | 0   | 0  |
| Абрамова О.А.    | 202      | 151    | 51        | 0                             | 0      | 0        | 0  | 20  | 0   | 0         | 0      | 51       | 0  | 0   | 40  | 0  |
| Антропова Н.Л.   | 71       | 0      | 71        | 0                             | 0      | 0        | 0  | 0   | 0   | 0         | 0      | 71       | 0  | 0   | 0   | 50 |
| Божедомова С.В.  | 164      | 164    | 0         | 0                             | 55     | 0        | 0  | 0   | 12  | 0         | 0      | 0        | 0  | 0   | 0   | 0  |
| Маркушина О.А.   | 42       | 42     | 0         | 0                             | 42     | 0        | 22 | 0   | 0   | 0         | 0      | 0        | 0  | 0   | 0   | 0  |
| Мастерова Л.З.   | 0        | 0      | 0         | 0                             | 0      | 0        | 0  | 0   | 0   | 0         | 0      | 0        | 0  | 0   | 0   | 0  |

Рисунок 34 - Экспорт данных о сводной нагрузке

### 3.7 Тестирование программного проекта

Во время отладки веб-приложения были устранены обнаруженные ошибки. Ниже приведено описание процесса их исправления.

Была допущена ошибка в имени пользователя, что привело к возникновению проблемы с подключением к базе данных. Скриншот выполнения веб-приложения с данной ошибкой показан на рисунке 35.

```
Warning: mysqli_connect(): (HY000/1045): Access denied for user 'home'@'localhost' (using password: NO) in C:\OpenServer\domains\tariffication\option.php on line 15
Невозможно соединиться с БД
```

Рисунок 35 - Выполнение web-приложения с ошибкой в имени пользователя

Исправленный вариант кода программы представлен на рисунке 36.

```
option.php
<?
$mode=$_COOKIE["mode"];

$fio=$_COOKIE["fio"];
$idtariffication=$_COOKIE["idtariffication"];

date_default_timezone_set("Europe/Moscow");
$date=date("Y")."-".date("m")."-".date("d");

$dblocation = "localhost";
$dbname = "tariffication";
$dbuser = "root";
$dbpasswd = "";

$dbcnx = mysqli_connect($dblocation,$dbuser,$dbpasswd,$dbname);

if(!$dbcnx)
{
?>
<meta charset="utf-8">
<?
echo 'Невозможно соединиться с БД';
exit;
}
?>
```

Рисунок 36 – Исправленный вариант кода программы

Также при отладки была обнаружена ошибка в имени базы данных. Скриншот выполнения web-приложения с ошибкой представлен на рисунке 37.

Warning: mysqli\_connect(): (HY000/1049): Unknown database 'tariffication' in C:\OpenServer\domains\tariffication\option.php on line 15  
Невозможно соединиться с БД

Рисунок 37 - Выполнение web-приложения с ошибкой в имени базы данных

Вместо tariffication в имени базы данных было указано tarification. Код программы с ошибкой в имени базы данных представлен на рисунке 38.

```
option.php
<?
smode=$_COOKIE["mode"];

sfiо=$_COOKIE["fiо"];
sidtariffication=$_COOKIE["idtariffication"];

date_default_timezone_set("Europe/Moscow");
sdate=date("Y")."-".date("m")."-".date("d");

sdblocation = "localhost";
sdbname = "tarification";
sdbuser = "root";
sdbpasswd = "";

sdbcnx = mysqli_connect(sdblocation,sdbuser,sdbpasswd,sdbname);

if(!sdbcnx)
{
?>
<meta charset="utf-8">
<?
echo 'Невозможно соединиться с БД';
exit;
}
?>
```

Рисунок 38 - Код программы с ошибкой в имени базы данных

Исправленный вариант кода программы представлен на рисунке 39.

```
option.php
<?
smode=$_COOKIE["mode"];

sfiо=$_COOKIE["fiо"];
sidtariffication=$_COOKIE["idtariffication"];

date_default_timezone_set("Europe/Moscow");
sdate=date("Y")."-".date("m")."-".date("d");

sdblocation = "localhost";
sdbname = "tariffication";
sdbuser = "root";
sdbpasswd = "";

sdbcnx = mysqli_connect(sdblocation,sdbuser,sdbpasswd,sdbname);

if(!sdbcnx)
{
?>
<meta charset="utf-8">
<?
echo 'Невозможно соединиться с БД';
exit;
}
?>
```

Рисунок 39 – Исправленный вариант кода программы

При отладке была выявлена ошибка проверки ФИО и контактных данных при добавлении нового преподавателя. После проверки с помощью javascript выводится предупреждение, происходит возврат на предыдущую страницу, но выполнение php-скрипта не прерывается, и преподаватель всё равно заносится в базу данных без ФИО, рисунок 40.

```
supd=$REQUEST["upd"];
sid=$REQUEST["id"];

steacher = $_POST["teacher"];
sphone = $_POST["phone"];
small = $_POST["mail"];

sidchair = $_POST["idchair"];

if (steacher=="")
{
?>
<meta http-equiv="content-type" content="text/html; charset=utf8" />
<script language="javascript">
alert ("Введите обязательные для ввода данные!");
history.back();
</script>
<?
}

if (supd==1)
{
mysql_query($dbcnx,"UPDATE teacher SET teacher='steacher', phone='sphone', mail='small' where idteacher=sid");
}
else
{//формирование SQL-запроса на добавление данных
mysql_query($dbcnx,"insert into teacher (teacher, phone, mail) values ('steacher', 'sphone', 'small')");
}
```

Рисунок 40 – Ошибка проверки ФИО преподавателя

Исправление ошибки путем добавления функции exit() в условия проверки представлено на рисунке 41.

```
supd=$REQUEST["upd"];
sid=$REQUEST["id"];

steacher = $_POST["teacher"];
sphone = $_POST["phone"];
small = $_POST["mail"];

sidchair = $_POST["idchair"];

if (steacher=="")
{
?>
<meta http-equiv="content-type" content="text/html; charset=utf8" />
<script language="javascript">
alert ("Введите обязательные для ввода данные!");
history.back();
</script>
<?
exit();
}

if (supd==1)
{
mysql_query($dbcnx,"UPDATE teacher SET teacher='steacher', phone='sphone', mail='small' where idteacher=sid");
}
else
{//формирование SQL-запроса на добавление данных
mysql_query($dbcnx,"insert into teacher (teacher, phone, mail) values ('steacher', 'sphone', 'small')");
}
```

Рисунок 41 – Исправление ошибки

При отладке была выявлена ошибка вывода перечня дисциплин. Результат выполнения web-приложения с ошибкой представлен на рисунке 42.

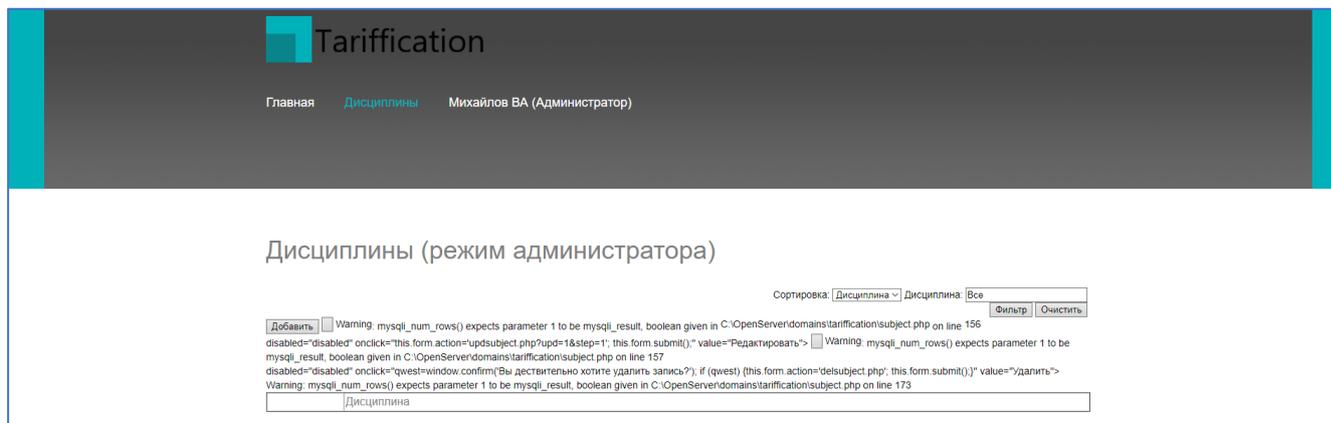


Рисунок 42 - Результат выполнения web-приложения с ошибкой

Причиной ошибки оказалось неверное имя таблицы дисциплин, рисунок 43.

```
<?
    $filter=$_GET["filter"];//считывание параметра фильтра
    $sort=$_GET["sort"];//считывание параметра фильтра

if ($filter==1)/*есть ли фильтрация данных*/
{
    $value1 = $_POST['FilterValue1'];//значение первого поля
    $ss="SELECT subject.* FROM subject where UPPER(subject) ";
    if ($value1!="Все")
    $ss=$ss." LIKE UPPER('%$value1.%') ";
    else
    $ss=$ss."=UPPER(subject) ";
}
else
    $ss="SELECT subject.* FROM subjects";

if ($sort==1)/*есть ли сортировка данных*/
{
    $fieldsort = $_POST['sortname'];//первое поле
    $ss=$ss." order by $fieldsort";
}
else
    $ss=$ss." order by subject";

    $r=mysql_query($dbcnx,$ss);
?>
```

Рисунок 43 - Неверное имя таблицы дисциплин

Для исправления ошибки было присвоено верное значение имени таблицы subject, рисунок 44.

```
subject.php
<?
    $filter=$_GET["filter"]; // считывание параметра фильтра
    $sort=$_GET["sort"]; // считывание параметра фильтра

if ($filter==1) /* есть ли фильтрация данных */
{
    $value1 = $_POST['FilterValue1']; // значение первого поля
    $ss="SELECT subject.* FROM subject where UPPER(subject)";
    if ($value1!="Все")
    $ss=$ss." LIKE UPPER('%$value1%' ) ";
    else
    $ss=$ss."=UPPER(subject) ";
}
else
    $ss="SELECT subject.* FROM subject ";

if ($sort==1) /* есть ли сортировка данных */
{
    $fieldsort = $_POST['sortname']; // первое поле
    $ss=$ss." order by $fieldsort";
}
else
    $ss=$ss." order by subject";

    $r=mysqli_query($dbcnx,$ss);
?>
```

Рисунок 44 – Исправление ошибки

Таким образом были исправлены все ошибки в работе web-приложения. Тестирование успешно завершено, web-приложение готово для использования.

### 3.8 Обоснование экономической эффективности проекта

Трудоемкость программного продукта – это мера усилий, которая требуется для его создания, которая измеряется в человеко/часах. Трудоемкость зависит от ряда факторов, таких как: размер программы, сложность функционала, опыт разработчика, используемый стек средств разработки и т.д.

Трудоемкость выполнения программного продукта важна для планирования бюджета и сроков выполнения проекта. Реалистичная оценка трудоемкости имеет большое значение для успешного выполнения программного проекта [4].

Перед подсчетом трудоемкости необходимо охарактеризовать основные этапы его создания. В ходе выполнения представленной работы были выполнены следующие этапы:

- анализ технико-экономической характеристики колледжа;
  - построение функциональной модели существующей организации;
  - построение функциональной модели улучшенной организации бизнес-процессов;
  - анализ имеющихся разработок;
  - проектирование диаграммы вариантов использования, состояний, деятельности, компонентов и развертывания;
  - проектирование модели базы данных;
  - разработка и тестирование программы;
  - описание контрольного примера реализации проекта;
  - подготовка руководства пользователя и руководства администратора.
- Определим трудоёмкость по видам работ и представим их в таблице 6.

Таблица 6 - Трудоемкость работ

| Виды работ  | Трудоёмкость |
|---|--------------|
| Анализ технико-экономической характеристики               | 5            |
| Построение функциональной модели существующей организации | 20           |
| Построение функциональной модели улучшенной организации   | 15           |
| Анализ существующих разработок                            | 10           |
| UML проектирование  | 25           |
| Проектирование инфологической модели базы данных          | 30           |
| Разработка и тестирование программы                       | 30           |
| Описание контрольного примера                             | 10           |
| Оформление документации                                   | 10           |
| Итого   | 155          |

Суммарная трудоемкость составила 155 чел./час.

Рассчитываем основную заработную плату. Заработная плата программиста без опыта работы составляет 20000 тыс. руб. в месяц. (21 рабочий день, 8-час. рабочий день) или 119 руб./час.

Таким образом, расходы на заработную плату по нашему проекту составляют:

$$ЗП = 119 \cdot 155 = 18452,38 \text{ руб.} \quad (1)$$

Отчисления на социальное страхование составляют 34%. В денежном выражении составляют:

$$18452,38 \cdot 0,34 = 6273,80 \text{ руб.} \quad (2)$$

Рассчитаем затраты на материалы.

Материалы, затраченные на создание проекта, приведены в таблице 7.

Таблица 7 - Расчет затрат на материалы

| Материальные ресурсы | Единицы измерения | Потребляемое количество | Цена за единицу, руб. | Сумма, руб. |
|----------------------|-------------------|-------------------------|-----------------------|-------------|
| Заправка картриджа   | шт.               | 1                       | 500                   | 500         |
| Бумага формата А4    | упаковка          | 1                       | 400                   | 400         |
| Папка                | шт.               | 1                       | 300                   | 300         |
| Флеш-накопитель      | шт.               | 1                       | 500                   | 500         |
| Итого                |                   |                         |                       | 1700        |

Рассчитаем амортизацию. Стоимость компьютера 50000 руб. Используем его в течении 6 лет (72 мес.). Годовая амортизация составит в месяц 694,44 руб. В час 0,96 руб. Умножив на трудоёмкость, определим:

$$АО = 0,96 \cdot 155 = 150,23 \text{ руб.} \quad (3)$$

Рассчитаем расходы на электроэнергию.

ПК в среднем употребляет 0,3 кВт/час.  $0,3 \cdot 155 = 46,5$  кВт. Примерная стоимость 1 кВт/час. около 4,68 руб.

Итого:

$$46,5 \cdot 4,68 = 217,62 \text{ руб.} \quad (4)$$

Итак, общая сумма затрат на разработку данного программного продукта равна 26 794 рублям. Следовательно, продавать его дешевле этой суммы невозможно, либо необходимо использовать тиражирование, чтобы обеспечить экономическую эффективность программы.

Созданный программный продукт обладает следующими преимуществами:

- удобство и мобильность в эксплуатации;
- интуитивно понятный интерфейс;
- структура база данных согласована с заказчиком;
- печатные формы вывода данных согласованы с заказчиком;
- бесплатные доработки программного продукта;
- бесплатное сопровождение и настройка;
- бесплатное обучение персонала.

Ориентировочная рыночная цена аналогичного программного продукт составляет 40000,00 руб. В качестве аналогов были рассмотрены следующие программные продукты «1С:Зарплата и кадры», «Тарифицирование в образовании ПАРУС-Бюджет 10», «АРМ Преподаватель» Цифровизация Плюс»

Аналоги имеют большой набор функций, который может оказаться избыточным для исследуемого колледжа. Среднерыночная стоимость аналогичных программных продуктов составляет около 40 000 рублей. Если реализовать наш продукт по цене 26 794 рубля и отказаться от приобретения готового решения, то можно заработать 13 206 рублей на каждой копии.

Выводы по главе 3

Третья глава посвящена практическому воплощению проекта и оценке его экономической эффективности. Здесь рассматриваются следующие аспекты:

Физическая база данных. Приводится пример физической базы данных, созданной на основании спроектированной ранее информационной модели. Это включает в себя реализацию таблиц, индексов, триггеров и других элементов, необходимых для корректного функционирования системы. Подробно описываются все объекты базы данных и их взаимодействие друг с другом, что позволяет понять, как данные будут храниться и обрабатываться в реальной системе.

Системная архитектура проекта. Проводится подробное описание и обоснование выбранного архитектурного решения для проекта. Обсуждаются различные варианты архитектуры, их преимущества и недостатки, а также причины выбора конкретного варианта. Определяются компоненты системы, их функции и способы взаимодействия между собой. Описываются принципы распределения нагрузки, обеспечение отказоустойчивости и масштабируемости системы.

Контрольный пример реализации проекта. Представлен контрольный пример, иллюстрирующий, как работает система в реальных условиях. Этот пример показывает, как система справляется с задачами учета тарификации преподавателей, начиная от ввода данных о рабочем времени до формирования отчетов и расчетов заработной платы. Приведены примеры интерфейсов пользователей, алгоритмов обработки данных и результатов выполнения запросов.

Результаты тестирования. Описаны результаты тестирования системы, включая функциональные тесты, нагрузочные тесты и тесты на устойчивость к ошибкам. Показано, насколько успешно система справляется со своими задачами, какие проблемы были выявлены в ходе тестирования и как они были устранены. Подтверждается соответствие системы предъявленным требованиям и стандартам качества.

Экономическая эффективность проекта. Рассматривается экономическая эффективность внедрения автоматизированной информационной системы тарификации преподавателей. Оцениваются затраты на разработку, внедрение и эксплуатацию системы, а также возможные экономические выгоды, которые она

принесет. Анализируются показатели окупаемости проекта, сроки возврата инвестиций и факторы, влияющие на эти показатели.

Определение затрат и окупаемости. Определяется общая сумма затрат на проект, включая стоимость оборудования, программного обеспечения, трудозатрат разработчиков и специалистов по внедрению. Рассчитывается срок окупаемости системы, исходя из предполагаемой экономии на трудовых ресурсах, повышении точности расчетов и снижении количества ошибок. Приводятся расчеты, подтверждающие выгодность проекта для колледжа.

Таким образом, третья глава завершает исследование, представляя готовую физическую базу данных, обосновывая выбранную архитектуру проекта, демонстрируя успешность его реализации через контрольный пример и тестирование, а также подтверждая экономическую целесообразность внедрения системы.

## Заключение

Тема представленной работы - разработка проекта автоматизированной системы учета тарификации преподавателей на примере ГАПОУ ТСПК Тольяттинского социального-педагогического колледжа.

Объектом исследования в данной работе является учет тарификации преподавателей.

В качестве предмета выпускной квалификационной работы выступил непосредственно процесс проектирования и разработки проекта автоматизированной системы учета тарификации преподавателей.

В результате выполнения работы были выполнены следующие задачи:

- выполнить анализ современных автоматизированных информационных систем (АИС) тарификации преподавателей;
- выбрать архитектуру АИС тарификации преподавателей;
- разработать прототип базы данных АИС тарификации преподавателей и проведение тестирования ее функциональности;
- выполнить проектирование пользовательского интерфейса АИС тарификации преподавателей;
- оценить экономическую эффективность от использования АИС тарификации преподавателей.

Плюсы внедрения автоматизированной информационной системы тарификации преподавателей колледжа:

- исключение необходимости для руководителей заниматься трудоемкими процедурами по созданию учетных записей преподавателей и сбору справочной информации.
- повышение уровня объективности при контроле и мониторинге качества работы преподавателей.
- уменьшение объема бумажного документооборота и сокращение времени, затрачиваемого на обработку документов.

- формирование комплексной и достоверной оценки индивидуальной результативности преподавателей и оптимального использования кадровых ресурсов колледжа.

- прозрачная и точная оценка деятельности колледжа.

- предоставление преподавателям возможности грамотного планирования своей деятельности, заполнения индивидуальных планов и оперативной отчётности.

Практическая значимость автоматизированной информационной системы тарификации преподавателей колледжа заключается в следующем:

- целостность и надежность хранения оперативных и справочных данных о преподавательской деятельности колледжа.

- автоматизация процесса составления тарификационных листов и сводной ведомости.

- систематизация работы по распределению нагрузки между преподавателями.

- сортировка, фильтр и вывод информации в виде пакета документов (экспорт данных в excel).

Таким образом все поставленные задачи были успешно выполнены, а автоматизированная система учета тарификации преподавателей готова к внедрению в колледже.

## Список используемой литературы

1. Астахова И. Ф., Толстобров А. П., Мельников В. М. SQL в примерах и задачах: Учеб. пособие.- Мн.: Новое знание, 2022.
2. Балдин, К. В. Информационные системы в экономике: Учебное пособие / К.В. Балдин. – Москва : НИЦ ИНФРА-М, 2019. – 218 с.
3. Блиновская, Я. Ю. Введение в проектирование систем автоматизации: Учебное пособие / Я.Ю. Блиновская, Д.С. Задоя. – Москва : Форум, НИЦ ИНФРА-М, 2019. – 112 с. 24. Бодров, О. А.
4. Буч, Г. UML: Руководство пользователя. / Г. Буч, Джекобсон И. и др. - М.: ДМК, 2019 г. – 356 с.
5. Предметно-ориентированные экономические информационные системы: Учебник для вузов / О.А. Бодров. – М.: Гор. линия-Телеком, 2013. – 244 с.
6. Варфоломеева, А. О. Информационные системы предприятия: Учебное пособие / А. О. Варфоломеева, А.В. Коряковский, В.П. Романов. – Москва : НИЦ ИНФРА-М, 2019. – 283 с.
7. Васильков, А. В. Информационные системы и их безопасность: Учебное пособие / А. В. Васильков, А.А. Васильков, И.А. Васильков. – Москва : Форум, 2018. – 528 с.
8. Вдовин, В. М. Предметно-ориентированные экономические информационные системы: Учебное пособие / В.М. Вдовин. – Москва : Дашков и К, 2013. – 388 с. Гришин, А. В. Промышленные информационные системы и сети: практическое руководство / А.В. Гришин. – Москва : Радио и связь, 2019. – 176 с.
9. Вендров А.М. Проектирование программного обеспечения экономических информационных систем. – М.: Финансы и статистика, 2019.
10. Вендров, А.М. CASE-технологии. Современные методы и средства проектирования информационных систем. / А.М. Вендров. - М.: Финансы и статистика, 2019 г. – 758 с.

11. Глушаков С. В., Ломотько Д. В. Базы данных, 2018. 415 с.
12. Голицына О. Л., Максимов Н. В., Попов И. И. Базы данных: Учебное пособие. – М.: ФОРУМ: ИНФРА-М, 2019. 268 с.
13. Горбенко, А. О. Информационные системы в экономике "Лаборатория знаний", 2019 – 295 с. 46. Симионов, Ю. Ф.
14. Информационные технологии в управлении предприятием. Антология. – М.: Три квадрата, 2021.
15. Исаев, Г. Н. Информационные системы в экономике: Учебник для студентов вузов / Г.Н. Исаев. – Москва : Омега-Л, 2023. – 462 с.
16. Исаев, Г. Н. Информационные технологии: Учебное пособие / Г.Н. Исаев. – Москва : Омега-Л, 2023. – 464 с.
17. Липаев, В.В. Системное проектирование сложных программных средств для информационных систем. / В.В. Липаев. - М.: Синтег, 2019 г. – 156 с.
18. Петров В.Н. Информационные системы. С-П: Питер, 2019.
19. Ульман Л. PHP и MySQL: создание сайтов, 2-е изд. : Пер. с англ. — М. : ООО “ И.Д. Вильямс”, 2015. — 544 с .
20. Фаулер, М. UML в кратком изложении. / М. Фаулер. - М.: Мир, 2019 г. – 204 с.
21. Bruce Silver Bpmn Method and Style: A Levels-Based Methodology for Bpm Process Modeling and Improvement Using Bpmn 2.0 - Cody-Cassidy Press – 2017.
22. Korotkevitch Dmitri Pro SQL Server Internals - M. Apress – 2019.
23. Unified Modeling Language (UML) Diagrams [Электронный ресурс]. URL: <https://www.geeksforgeeks.org/unified-modeling-language-uml-introduction/> (дата обращения: 05.10.2024).
24. Visual Paradigm Online [Электронный ресурс]. URL: <https://online.visual-paradigm.com/> (дата обращения: 05.10.2024).
25. What is FURPS+? [Электронный ресурс]. URL: <https://www.coetl.com/furps-plus/> (дата обращения: 05.10.2024).

Приложение А  
Код программы

```
<?
$s="SELECT tariffication.*, subject, teacher, squad.*,
if (disabilities=1, 'Да', 'Нет') as disabilities,
if (check3=1, 'Да', 'Нет') as check3,
if (check5=1, 'Да', 'Нет') as check5,
if (check10=1, 'Да', 'Нет') as check10,
if (check15=1, 'Да', 'Нет') as check15,
(advanced1+consultations1+intermediate1+home1+coursework1) as total1,
(advanced2+consultations2+intermediate2+home2+coursework2) as total2,
(advanced1+consultations1+intermediate1+home1+coursework1+advanced2+consultati
ons2+intermediate2+home2+coursework2) as total
FROM tariffication, subject, teacher, squad where squad.idsquad=tariffication.idsquad
and teacher.idteacher=tariffication.idteacher and
subject.idsubject=tariffication.idsubject" ;

$r=mysqli_query($dbcnx,$s);
?>

<table WIDTH=100% border="1" cellspacing=0 cellpadding=2 style="table-layout:
auto" >

<tr>
<td rowspan="3"><font color=white>&nbsp;</font></td>
```



## Продолжение Приложения А

```
</tr>
```

```
<?
```

```
for ($i=0;$i<mysql_num_rows($r);$i++)
```

```
{
```

```
    echo "<tr>";
```

```
    echo "
```

```
    <td> $f[subject]</td>
```

```
    <td> $f[squad] $f[category]<br>$f[level]<br>Количество:
```

```
    $f[countstudent]</td>
```

```
    <td> $f[teacher]</td>
```

```
    <td> $f[disabilities] </td>
```

```
    <td> $f[check3] </td>
```

```
    <td> $f[check5] </td>
```

```
    <td> $f[check10] </td>
```

```
    <td> $f[check15] </td>
```

```
    <td> $f[advanced1] </td>
```

```
    <td> $f[advanced2] </td>
```

```
    <td> $f[consultations1] </td>
```

```
    <td> $f[consultations2] </td>
```

## Продолжение Приложения А

<td> \$f[intermediate1] </td>

<td> \$f[intermediate2] </td>

<td> \$f[home1] </td>

<td> \$f[home2] </td>

<td> \$f[coursework1] </td>

<td> \$f[coursework2] </td>

<td> \$f[total1] </td>

<td> \$f[total2] </td>

<td> \$f[total] </td>

</tr>";

}

?>

</table>

</form>

<?

\$s="SELECT teacher.\*,

(SELECT

IFNULL(SUM(advanced1+consultations1+intermediate1+home1+coursework1+advanced2+consultations2+intermediate2+home2+coursework2), 0) from tariffication where tariffication.idteacher=teacher.idteacher) as total,

(SELECT

IFNULL(SUM(advanced1+consultations1+intermediate1+home1+coursework1+advanced2+consultations2+intermediate2+home2+coursework2), 0) from tariffication, squad

## Продолжение Приложения А

where `tariffication.idteacher=teacher.idteacher` and `tariffication.idsquad=squad.idsquad` and category like 'Коммерческая') as budget,

(SELECT

IFNULL(SUM(advanced1+consultations1+intermediate1+home1+coursework1+advanced2+consultations2+intermediate2+home2+coursework2), 0) from tariffication, squad where `tariffication.idteacher=teacher.idteacher` and `tariffication.idsquad=squad.idsquad` and category like 'Некоммерческая') as nonbudget,

(SELECT

IFNULL(SUM(advanced1+consultations1+intermediate1+home1+coursework1+advanced2+consultations2+intermediate2+home2+coursework2), 0) from tariffication, squad where `tariffication.idteacher=teacher.idteacher` and `tariffication.idsquad=squad.idsquad` and `disabilities=1` and category like 'Коммерческая') as budgetdisabilities,

(SELECT

IFNULL(SUM(advanced1+consultations1+intermediate1+home1+coursework1+advanced2+consultations2+intermediate2+home2+coursework2), 0) from tariffication, squad where `tariffication.idteacher=teacher.idteacher` and `tariffication.idsquad=squad.idsquad` and `disabilities=1` and category like 'Некоммерческая') as nonbudgetdisabilities,

(SELECT

IFNULL(SUM(advanced1+consultations1+intermediate1+home1+coursework1+advanced2+consultations2+intermediate2+home2+coursework2), 0) from tariffication, squad where `tariffication.idteacher=teacher.idteacher` and `tariffication.idsquad=squad.idsquad` and category like 'Коммерческая' and level like 'Углубленная') as budgetadvanced,

(SELECT

IFNULL(SUM(advanced1+consultations1+intermediate1+home1+coursework1+advanced2+consultations2+intermediate2+home2+coursework2), 0) from tariffication, squad where `tariffication.idteacher=teacher.idteacher` and `tariffication.idsquad=squad.idsquad`

## Продолжение Приложения А

and category like 'Некоммерческая' and level like 'Углубленная') as  
nonbudgetadvanced,

(SELECT IFNULL(SUM(advanced1+advanced2), 0) from tariffication, squad where  
tariffication.idteacher=teacher.idteacher and tariffication.idsquad=squad.idsquad and  
check3=1 and category like 'Коммерческая') as budgetcheck3,

(SELECT IFNULL(SUM(advanced1+advanced2), 0) from tariffication, squad where  
tariffication.idteacher=teacher.idteacher and tariffication.idsquad=squad.idsquad and  
check3=1 and category like 'Некоммерческая') as nonbudgetcheck3,

(SELECT IFNULL(SUM(advanced1+advanced2), 0) from tariffication, squad where  
tariffication.idteacher=teacher.idteacher and tariffication.idsquad=squad.idsquad and  
check5=1 and category like 'Коммерческая') as budgetcheck5,

(SELECT IFNULL(SUM(advanced1+advanced2), 0) from tariffication, squad where  
tariffication.idteacher=teacher.idteacher and tariffication.idsquad=squad.idsquad and  
check5=1 and category like 'Некоммерческая') as nonbudgetcheck5,

(SELECT IFNULL(SUM(advanced1+advanced2), 0) from tariffication, squad where  
tariffication.idteacher=teacher.idteacher and tariffication.idsquad=squad.idsquad and  
check10=1 and category like 'Коммерческая') as budgetcheck10,

(SELECT IFNULL(SUM(advanced1+advanced2), 0) from tariffication, squad where  
tariffication.idteacher=teacher.idteacher and tariffication.idsquad=squad.idsquad and  
check10=1 and category like 'Некоммерческая') as nonbudgetcheck10,

(SELECT IFNULL(SUM(advanced1+advanced2), 0) from tariffication, squad where  
tariffication.idteacher=teacher.idteacher and tariffication.idsquad=squad.idsquad and  
check15=1 and category like 'Коммерческая') as budgetcheck15,

(SELECT IFNULL(SUM(advanced1+advanced2), 0) from tariffication, squad where  
tariffication.idteacher=teacher.idteacher and tariffication.idsquad=squad.idsquad and  
check15=1 and category like 'Некоммерческая') as nonbudgetcheck15

FROM teacher ";

## Продолжение Приложения А

```
$r=mysqli_query($dbcnx,$s);
```

```
?>
```

```
<table WIDTH=100% border="1" cellspacing=0 cellpadding=2 >
```

```
<tr>
```

```
<td rowspan="4">
```

```
<h1>Преподаватель</h1>
```

```
</td>
```

```
<td colspan="3">
```

```
<h1>Нагрузка</h1>
```

```
</td>
```

```
<td colspan="12">
```

```
<h1>Доплаты (от количества часов)</h1>
```

```
</td>
```

```
</tr>
```

```
<tr>
```

```
<td rowspan="3">
```

```
<h1>Всего</h1>
```

```
</td>
```

```
<td rowspan="3">
```

```
<h1>Бюджет</h1>
```

```
</td>
```

```
<td rowspan="3">
```

```
<h1>Внебюджет</h1>
```

```
</td>
```

## Продолжение Приложения А

<td colspan="6">

<h1>Бюджет</h1>

</td>

<td colspan="6">

<h1>Внебюджет</h1>

</td>

</tr>

<tr>

<td rowspan="2">

<h1>ОВЗ</h1>

</td>

<td rowspan="2">

<h1>Углубл</h1>

</td>

<td colspan="4">

<h1>Проверка тетрадей</h1>

</td>

<td rowspan="2">

<h1>ОВЗ</h1>

</td>

<td rowspan="2">

<h1>Углубл</h1>

</td>

<td colspan="4">

<h1>Проверка тетрадей</h1>

</td>

</tr>

<tr>

## Продолжение Приложения А

<td>

<?>

```
for ($i=0;$i<mysql_num_rows($r);$i++)  
{
```

```
    echo "<tr>";
```

```
    echo "
```

```
    <td> $f[total]</td>
```

```
    <td> $f[budget]</td>
```

```
    <td> $f[nonbudget]</td>
```

```
    <td> $f[budgetdisabilities]</td>
```

```
    <td> $f[budgetadvanced]</td>
```

```
    <td> $f[budgetcheck3]</td>
```

```
    <td> $f[budgetcheck5]</td>
```

```
    <td> $f[budgetcheck10]</td>
```

```
    <td> $f[budgetcheck15]</td>
```

```
    <td> $f[nonbudgetdisabilities]</td>
```

```
    <td> $f[nonbudgetadvanced]</td>
```

```
    <td> $f[nonbudgetcheck3]</td>
```

```
    <td> $f[nonbudgetcheck5]</td>
```

```
    <td> $f[nonbudgetcheck10]</td>
```

```
    <td> $f[nonbudgetcheck15]</td>
```

## Продолжение Приложения А

```
        ";
        echo "</tr>";
    }
?>

</table>

</form>

<?

if (isset ($_POST['Arrtariffication']))
{
    $Arr=$_POST['Arrtariffication'];
    $idtariffication=$Arr[0];
    setcookie ( 'idtariffication', $idtariffication);
}

$step=$_REQUEST["step"];
$upd=$_REQUEST["upd"];
$now=date("Y")."-".date("m")."-".date("d");

if ($step==2)
{

//считываем данные из формы
$id squad = $_POST["idsquad"];
```

## Продолжение Приложения А

```
$idsubject = $_POST["idsubject"];
$idteacher = $_POST["idteacher"];
$disabilities = $_POST["disabilities"];
$check3 = $_POST["check3"];
$check5 = $_POST["check5"];
$check10 = $_POST["check10"];
$advanced1 = $_POST["advanced1"];
$advanced2 = $_POST["advanced2"];
$consultations1 = $_POST["consultations1"];
$consultations2 = $_POST["consultations2"];
$intermediate1 = $_POST["intermediate1"];
$intermediate2 = $_POST["intermediate2"];
$home1 = $_POST["home1"];
$home2 = $_POST["home2"];
$coursework1 = $_POST["coursework1"];
$coursework2 = $_POST["coursework2"];
if ($upd==1)
{
    mysqli_query($dbcnx,"UPDATE tariffication SET idsquad='$idsquad',
idsubject='$idsubject', idteacher='$idteacher', disabilities='$disabilities',
check3='$check3', check5='$check5', check10='$check10', advanced1='$advanced1',
advanced2='$advanced2', consultations1='$consultations1',
consultations2='$consultations2', intermediate1='$intermediate1',
intermediate2='$intermediate2', home1='$home1', home2='$home2',
coursework1='$coursework1', coursework2='$coursework2' where
idtariffication=$idtariffication");
}
else
```

## Продолжение Приложения А

```
{
    mysqli_query($dbcnx,"INSERT INTO tariffication ( idsquad, idsubject,
idteacher, disabilities, check3, check5, check10, advanced1, advanced2, consultations1,
consultations2, intermediate1, intermediate2, home1, home2, coursework1,
coursework2) VALUES ('$idsquad', '$idsubject', '$idteacher', '$disabilities', '$check3',

'$check5', '$check10', '$advanced1', '$advanced2', '$consultations1', '$consultations2',
'$intermediate1', '$intermediate2', '$home1', '$home2', '$coursework1',
'$coursework2')");
}
?>
<script language="javascript">
location.href='tariffication.php?filter=0';
</script>
<?
}
?>

<table width="622" border="0">
<tr>
<tr>
<td>Работа с обучающимися с ОВЗ:</td>
<td><input type="checkbox" <? if ($f["disabilities"]) echo "checked";?> value=1
name="disabilities"></td>
</tr>
```

## Продолжение Приложения А

```
<tr>
<td>Проверка письменных работ 3%:</td>
<td><input type="checkbox" <? if ($f["check3"]) echo "checked";?> value=1
name="check3"></td>
</tr>
```

```
<tr>
<td>Проверка письменных работ 5%:</td>
<td><input type="checkbox" <? if ($f["check5"]) echo "checked";?> value=1
name="check5"></td>
</tr>
```

```
<tr>
<td>Проверка письменных работ 10%:</td>
<td><input type="checkbox" <? if ($f["check10"]) echo "checked";?> value=1
name="check10"></td>
</tr>
```

```
<tr>
<td>Проверка письменных работ 15%:</td>
<td><input type="checkbox" <? if ($f["check15"]) echo "checked";?> value=1
name="check15"></td>
</tr>
```

```
<tr>
<td>Проверка письменных работ 15%:</td>
<td><input type="checkbox" <? if ($f["check15"]) echo "checked";?> value=1
name="check15"></td>
```

## Продолжение Приложения А

</tr>

<tr>

<td>Проверка письменных работ 15%:</td>

<td><input type="checkbox" <? if (\$f["check15"]) echo "checked";?> value=1  
name="check15"></td>

</tr>

<tr>

<td>Проверка письменных работ 15%:</td>

<td><input type="checkbox" <? if (\$f["check15"]) echo "checked";?> value=1  
name="check15"></td>

</tr>

<tr>

<td>Часы по УП (1 семестр):</td>

<td><input type="checkbox" <? if (\$f["check15"]) echo "checked";?> value=1  
name="check15"></td>

</tr>

<tr>

<td>Часы по УП (2 семестр):</td>

<td><input type="checkbox" <? if (\$f["check15"]) echo "checked";?> value=1  
name="check15"></td>

</tr>

<tr>

## Продолжение Приложения А

```
<td>Консультации (1 семестр):</td>
<td><input type="checkbox" <? if ($f["check15"]) echo "checked";?> value=1
name="check15"></td>
</tr>

<tr>
<td>Консультации (2 семестр):</td>
<td><input type="checkbox" <? if ($f["check15"]) echo "checked";?> value=1
name="check15"></td>
</tr>

<tr>
<td>Промежуточная аттестация (1 семестр):</td>
<td><input type="checkbox" <? if ($f["check15"]) echo "checked";?> value=1
name="check15"></td>
</tr>

<tr>
<td>Промежуточная аттестация (2 семестр):</td>
<td><input type="checkbox" <? if ($f["check15"]) echo "checked";?> value=1
name="check15"></td>
</tr>

<tr>
<td>Домашние контрольные работы (1 семестр):</td>
<td><input type="checkbox" <? if ($f["check15"]) echo "checked";?> value=1
name="check15"></td>
</tr>
```