

Федеральное государственное автономное
образовательное учреждение
высшего образования
«СИБИРСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

Хакасский технический институт – филиал ФГАОУ ВО
«Сибирский федеральный университет»

Кафедра прикладной информатики, естественно-научных и гуманитарных
дисциплин

УТВЕРЖДАЮ
Заведующий кафедрой
_____ О. В. Папина
подпись
«_____» _____ 2023 г.

БАКАЛАВРСКАЯ РАБОТА

09.03.03 – Прикладная информатика

Автоматизированное рабочее место специалиста по документационному
обеспечению кафедры ХТИ – филиала СФУ

Руководитель _____ доцент, канд. физ-мат. наук М. А. Буреева
подпись, дата

Выпускник _____ Р. Г. Гусев
подпись, дата

Консультанты
по разделам:

Экономический _____ М. А. Буреева
подпись, дата

Нормоконтролер _____ А. Н. Кадычегова
подпись, дата

Федеральное государственное автономное
образовательное учреждение
высшего образования
«СИБИРСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

Хакасский технический институт – филиал ФГАОУ ВО
«Сибирский федеральный университет»

Кафедра прикладной информатики, естественно-научных и гуманитарных
дисциплин

УТВЕРЖДАЮ
Заведующий кафедрой

_____ О. В. Папина

подпись

«_____» _____ 2023 г.

**ЗАДАНИЕ
НА ВЫПУСКНУЮ КВАЛИФИКАЦИОННУЮ РАБОТУ
в форме бакалаврской работы**

Студенту Гусеву Роману Георгиевичу

Группа ХБ 19-02

Направление 09.03.03 Прикладная информатика

Тема выпускной квалификационной работы: Автоматизированное рабочее место специалиста по документационному обеспечению кафедры ХТИ – филиала СФУ.

Утверждена приказом по институту № 283 от 11.05.2023 г.

Руководитель ВКР: М. А. Буреева, доцент, канд. физ.-мат. наук, ХТИ – филиал СФУ

Исходные данные для ВКР: заказ ХТИ – филиала СФУ, шаблоны документов по кафедре ХТИ – филиала СФУ.

Перечень разделов ВКР:

1. Анализ предметной области деятельности кафедры ПИЕиГД ХТИ – филиала СФУ.

2. Разработка АРМа специалиста по документационному обеспечению кафедры ПИЕиГД ХТИ – филиала СФУ.

3. Расчет экономических затрат на реализацию АРМа специалиста по документационному обеспечению кафедры ПИЕиГД ХТИ – филиала СФУ.

Руководитель ВКР

М. А. Буреева

подпись, дата

Задание принял к исполнению

Р. Г. Гусев

подпись, дата

«11» мая 2023 г.

РЕФЕРАТ

Выпускная квалификационная работа по теме «Автоматизированное рабочее место специалиста по документационному обеспечению кафедры ХТИ – филиала СФУ» содержит 64 страницы текстового документа, 21 рисунок, 17 таблиц, 8 формул, 13 использованных источников.

АВТОМАТИЗИРОВАННОЕ РАБОЧЕЕ МЕСТО, ПРИЛОЖЕНИЕ, ОТЧЕТ, РИСКИ, ПРОЕКТИРОВАНИЕ, СИСТЕМА, IDEF, DFD, БАЗА ДАННЫХ, PYTHON.

Объект ВКР: процесс автоматизации выполнения функциональных обязанностей специалистом по документационному обеспечению кафедры ХТИ – филиала СФУ.

Предмет ВКР: автоматизированное рабочее место специалиста по документационному обеспечению ХТИ – филиала СФУ

Цель работы: создание автоматизированного рабочего места специалиста кафедры для улучшения качества его работы.

В ходе работы были выполнены следующие задачи:

- Проанализирована деятельность ХТИ – филиала СФУ.
- Выполнен сравнительный анализ АРМ, работающих в похожей сфере деятельности.
- Выбраны средства разработки.
- Разработан АРМ специалиста по документационному обеспечению кафедры ХТИ – филиала СФУ.
- Рассчитаны затраты и определены риски, которым подвержен проект.

После анализа деятельности специалиста кафедры выяснилось, что его задачи требуют много времени и включают в себя множество рутинных операций. В результате этого было разработано автоматизированное рабочее место специалиста кафедры по документационному обеспечению, которое позволит улучшить качество его работы.

SUMMARY

The theme of the graduation thesis is «Automated Workplace Development for a Department Specialist in Documentation Support at Khakass Technical Institute – branch of Siberian Federal University». It consists of 64 pages, 21 figures, 17 tables, 8 formulae, 13 references items.

AUTOMATED WORKPLACE, APPLICATION, REPORT, RISKS, DESIGNING, SYSTEM, IDEF, DFD, DATABASE, ESTIMATE, PYTHON.

Object of study – an automated workplace development for a department specialist in documentation support at KhTI – branch of SibFU.

Purpose of research: improving the quality of work of a department specialist in documentation support by designing an automated workplace.

Objectives to be considered:

- to analyze the workforce of the KhTI – branch of SibFU;
- to perform a comparative analysis of automated workplaces implemented in a similar field of activity;
- to select development tools;
- the automated workplace has been developed;
- to calculate costs and economic benefits of the project, and to determine risks to which the project is exposed.

Results: the analysis of workflow of the department specialist in documentation support has revealed routine and time-consuming aspects of his/her work.

Thus, the automated workplace for a department specialist in documentation support at KhTI – branch of SibFU has been developed. This will allow improving the quality of the specialist's work.

English language supervisor

N.V. Chezybaeva

ОГЛАВЛЕНИЕ

Введение	6
1 Анализ предметной области и деятельности кафедры ПИЕиГД ХТИ – филиала СФУ	8
1.1 Организационная характеристика предметной области	8
1.2 Организационная характеристика кафедры «Прикладная информатика, естественно-научные и гуманитарные дисциплины»	10
1.3 Анализ деятельности кафедры ПИЕиГД	12
1.4 Функциональная структура кафедры ПИЕиГД как объекта автоматизации рабочего места	12
1.5 Информационная модель работы специалиста по документационному обеспечению кафедры	14
1.6 Обоснование необходимости автоматизации рабочего места специалиста по документационному обеспечению кафедры	17
1.7 Требования к АРМ специалиста по документационному обеспечению кафедры. Постановка цели и задач	18
1.8 Сравнительный анализ АРМ, работающих в похожей сфере деятельности	19
Выводы по разделу «Анализ предметной области и деятельности кафедры ПИЕиГД ХТИ – филиала СФУ»	20
2 Разработка АРМа специалиста по документационному обеспечению кафедры ПИЕиГД ХТИ – филиала СФУ	20
2.1 Модель новой организации процесса ТО-ВЕ	20
2.2 Диаграмма потоков данных	23
2.3 Проектирование базы данных. Разработка модели «сущность-связь»	25
2.4 Выбор программных средств для разработки	29
2.5 Пользовательский интерфейс АРМа	30
2.6 Описание интерфейса АРМа и программных модулей	34
2.6.1 Работа с БД	35

2.6.2	Метод для создания элементов в списке	41
2.6.3	Редактор документов	43
	Выводы по разделу «Разработка АРМа специалиста по документационному обеспечению кафедры ПИЕиГД ХТИ – филиала СФУ»	46
3	Расчет экономических затрат на реализацию АРМа специалиста по документационному обеспечению кафедры ПИЕиГД ХТИ – филиала СФУ	47
3.1	Методика оценки экономической эффективности разработки АРМа	47
3.2	Капитальные затраты	48
3.3	Эксплуатационные затраты	53
3.4	Расчет затрат реализации проекта. МетодикаТСО (Total Cost of Ownership)	55
3.5	Риски реализации проекта	57
	Выводы по разделу «Расчет экономических затрат на реализацию АРМа специалиста по документационному обеспечению кафедры ПИЕиГД ХТИ – филиала СФУ»	58
	Заключение	60
	Список использованных источников	62

ВВЕДЕНИЕ

В выпускной квалификационной работе рассматривается объект исследования – рабочее место специалиста по документационному обеспечению кафедры ПИЕиГД ХТИ – филиала СФУ и процесс автоматизации выполнения его функциональных обязанностей.

Цель работы: улучшение качества выполнения работы специалистом кафедры ПИЕиГД ХТИ путем разработки автоматизированного рабочего места.

Для достижения цели необходимо решить следующие задачи:

- Проанализировать деятельность ХТИ – филиала СФУ.
- Выполнить сравнительный анализ АРМ, работающих в похожей сфере деятельности. Поставить цели и задачи.
- Выбрать средства разработки.
- Разработать АРМ специалиста кафедры (модуль научно-педагогической деятельности преподавателей).
- Рассчитать затраты и экономическую выгоду проекта, определить риски, которым подвержен проект.

Анализ работы специалиста кафедры показал, что текущая организация обработки документов весьма затратная по времени, в связи с чем для работника кафедры является неэффективной. В результате этого выявления возникла необходимость разработки автоматизированного рабочего места специалиста по документационному обеспечению кафедры ХТИ – филиала СФУ (модуль научно-педагогической деятельности преподавателей).

В итоге было решено разработать программный продукт, позволяющий улучшить качество работы с документными данными, что позволит быстро обрабатывать информацию, снизить трудоемкость и повысить качество хранения, оперативность получения результатной информации и итоговых документов.

В выпускной квалификационной работе будет разработано прикладное решение, в котором будут включены необходимые функции для ведения отчетности о научной деятельности профессорско-преподавательского состава кафедры ХТИ – филиала СФУ.

В первом разделе рассматривается деятельность ХТИ – филиала СФУ, построена модель потоков данных, где представлена информация, каким образом происходит заполнение документации о научной деятельности профессорско-преподавательского состава кафедры, а также даются обоснования для разработки АРМа.

Второй раздел представляет описание разработки АРМа.

В основе третьего раздела выпускной квалификационной работы лежит анализ единовременных затрат для разработки АРМ специалиста и обоснование экономического эффекта от внедрения АРМа на кафедру ХТИ – филиала СФУ.

1 Анализ предметной области и деятельности кафедры ПИЕиГД ХТИ – филиала СФУ

1.1 Организационная характеристика предметной области

В Хакасском техническом институте – филиале федерального государственного автономного образовательного учреждения высшего образования «Сибирский федеральный университет» функционирует множество информационных систем, предназначенных для реализации образовательного процесса. Однако в обязанности каждого преподавателя ХТИ – филиала СФУ, входит и выполнение научно-исследовательской деятельности. За отчетность по данному виду работы на каждой кафедре отвечает специалист по документационному обеспечению. Поэтому в ХТИ – филиале СФУ возникла потребность в разработке АРМа специалист по документационному обеспечению кафедры (модуль отчетности о научной деятельности профессорско-преподавательского состава кафедры).

Краткое наименование организации: ХТИ – филиал СФУ.

Юридический адрес учреждения: Республика Хакасия, г. Абакан, ул. Щетинкина, 27 (корпус «А»), 655017.

Контактный телефон: (8 3902) 22-53-55.

Основной задачей ХТИ – филиала СФУ является создание передовой образовательной, научно-исследовательской и инновационной инфраструктуры, продвижение новых знаний и технологий для решения задач социально-экономического развития Сибирского федерального округа, а также создание кадрового потенциала — конкурентоспособных специалистов по приоритетным направлениям развития Сибири и Российской Федерации, соответствующих современным интеллектуальным требованиям и отвечающих мировым стандартам [12].

ХТИ – филиал СФУ имеет лицензию на осуществление образовательной деятельности, выданную Федеральной службой по надзору в сфере образования и науки 4 июля 2016 г. (бессрочно), регистрационный номер № Л035-00115-24/00096964, Свидетельство о государственной аккредитации серии 90А01 № 0003102, выданное Федеральной службой по надзору в сфере образования и науки на срок до 19 декабря 2024 г., рег. № 2957 [12].

Институт имеет сертификат соответствия системы менеджмента качества требованиям ГОСТ ISO 9001-2011 (ISO 9001:2008) применительно к деятельности в области образования [12].

В ХТИ – филиале СФУ ведутся научно-исследовательские разработки, тематика которых ориентирована на нужды Республики Хакасия. Перечень инвестиционных проектов, разработанных студентами по программам социально-экономического развития муниципальных образований РХ довольно велик, и тематика разнообразна.

Организационная структура ХТИ – филиала СФУ представлена на рисунке 1.1.

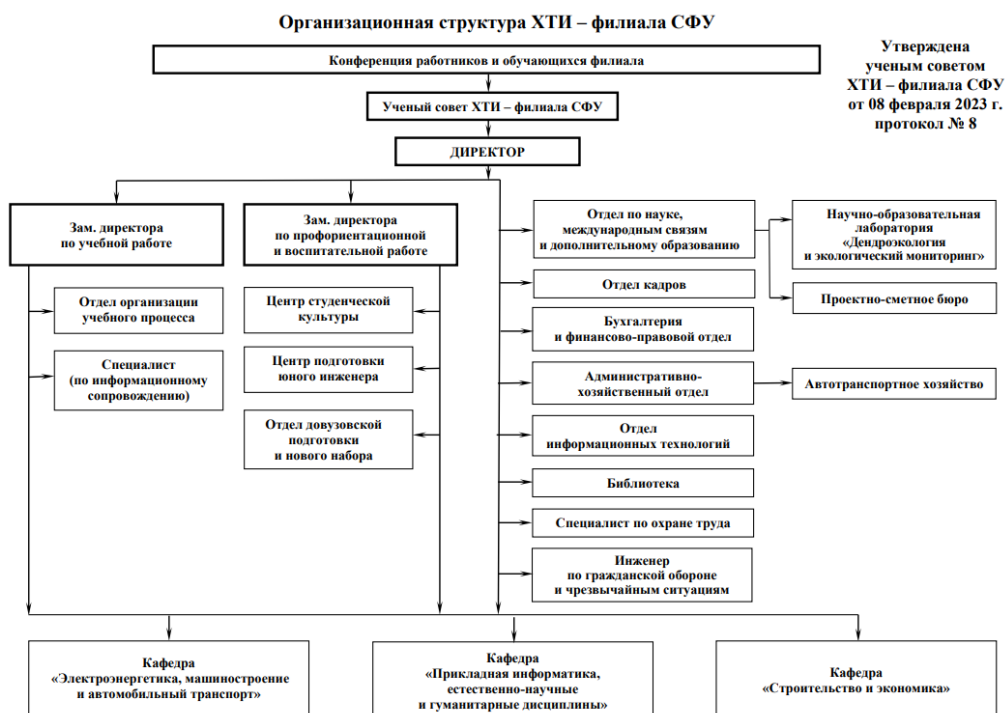


Рисунок 1.1 – Организационная структура ХТИ – филиала СФУ

1.2 Организационная характеристика кафедры «Прикладная информатика, естественно-научные и гуманитарные дисциплины»

Кафедра «Прикладная информатика, естественно-научные и гуманитарные дисциплины» (далее – кафедра, кафедра ПИЕиГД) является структурным подразделением Хакасского технического института – филиала федерального государственного автономного образовательного учреждения высшего образования «Сибирский федеральный университет» (далее – институт, ХТИ – филиал СФУ)

Кафедра создана приказом директора на основании решения ученого совета института от 1 сентября 2022 года (протокол №1), путем реорганизации кафедры «Прикладная информатика, математика и естественно-научные» дисциплины» и кафедры «Экономика и гуманитарные дисциплины»

В своей учебной деятельности кафедра осуществляет учебную, учебно-методическую и научно-исследовательскую деятельность. Совместно с другими подразделениями института ведет подготовку студентов, осуществляет воспитательную работу среди студентов. Кафедра ПИЕиГД является выпускающей. Кафедра ПИЕиГД реализует основную образовательную программу по направлению подготовки 09.03.03 Прикладная информатика (уровень бакалавриата).

Кафедра частично реализует основные образовательные программы высшего образования по направлениям подготовки бакалавриата/специалитета:

- 08.03.01 Строительство,
- 13.03.02 Электроэнергетика и электротехника,
- 15.03.05 Конструкторско-технологическое обеспечение машиностроительных производств,
- 23.03.03 Эксплуатация транспортно-технологических машин и комплексов,
- 38.03.01 Экономика,

- 08.05.01 Строительство уникальных зданий и сооружений,
Частично реализует образовательные программы высшего образования по направлениям магистратуры

- 08.04.01 Строительство,

- 13.04.02 Электроэнергетика и электротехника.

Месторасположение кафедры ПИЕиГД: Республика Хакасия, г. Абакан, ул. Щетинкина, 27 (корпус «А»), каб. 203.

Телефон: +7(913)544-87-74.

E-mail: mied2011@mail.ru.

Кадровый состав кафедры определяется штатным расписанием, которое утверждается приказом директора института.

Штатный состав кафедры включает следующих сотрудников:

Профессорско-преподавательский состав:

- заведующий кафедрой;
- профессор;
- доцент;
- старший преподаватель;
- ассистент.

Учебно-вспомогательный персонал:

- инженер;
- лаборант;
- специалист по документационному обеспечению.

Кадровый состав кафедры определяется штатным расписанием, которое утверждается приказом директора института.

Руководство работой кафедры ПИЕиГД осуществляет заведующий кафедрой, непосредственно подчиняющийся директору.

1.3 Анализ деятельности кафедры ПИЕиГД

В ходе анализа деятельности кафедры ПИЕиГД было выяснено, что он выполняет следующие задачи:

1) осуществляет меры по защите информации в соответствии с требованиями нормативных документов;

2) проводит все виды учебных занятий по всем формам обучения по закрепленным за кафедрой дисциплинам в соответствии с учебным планом, рабочими программами учебных дисциплин, расписанием учебных занятий и требованиями локальных актов;

3) участвует в мероприятиях, проводимых институтом, по совершенствованию учебного и научного процессов и повышению качества обучения;

4) проводит научные исследования по приоритетным направлениям, определенными программой развития института;

5) проводит научно-методические семинары с обсуждением вопросов, направленных на повышение уровня педагогического мастерства профессорско-преподавательского состава, заслушивает отчеты преподавателей о повышении квалификации.

1.4 Функциональная структура кафедры ПИЕиГД как объекта автоматизации рабочего места

Объектом автоматизации выбрано рабочее место специалиста по документационному обеспечению кафедры ПИЕиГД ХТИ – филиала СФУ, выполняющего свою деятельность на основании Должностной инструкции специалиста кафедры по документационному обеспечению кафедры ПИЕиГД ХТИ - филиала СФУ, уставом ФГАОУ ВПО «Сибирский федеральный университет», положением о ХТИ – филиале СФУ.

В ходе своей профессиональной деятельности специалист по документационному обеспечению кафедры выполняет следующие функции:

1. Осуществляет подготовку документов для заведующего кафедрой.
2. Обеспечивает учет и регистрацию, входящих на кафедру документов, информационно-справочное обслуживание и хранение документов.
3. Контролирует качество подготовки, правильность составления, согласования, утверждения документов, представляемых на подпись заведующему кафедрой.
4. Осуществляет контроль исполнения документов и поручений заведующего кафедрой, принимать оперативные меры, направленные на своевременное и качественное их выполнение.
5. Выполняет работу по документационному обеспечению деятельности кафедры, в т. ч. ведет табель учета рабочего времени учебно-вспомогательного персонала кафедры.
6. Обеспечивает выполнение машинописных и копировально-множительных работ.
7. Проводит ежегодный отбор документов на архивное хранение или на уничтожение.
8. Осуществляет подготовку проводимых на кафедре совещаний, переговоров и иных мероприятий.
9. Соблюдает правила эксплуатации технических средств обработки передачи информации.

Одним из самых трудоемких видов деятельности специалиста является формирование отчетности о научной деятельности профессорско-преподавательского состава кафедры.

1.5 Информационная модель работы специалиста по документационному обеспечению кафедры

В связи с тем, что у специалиста кафедры монотонные и трудные работы по обработке документационных данных, появилась необходимость в автоматизации этих работ.

Информационная модель существующей системы до автоматизации представлена на рисунке 1.2.



Рисунок 1.2 – Информационная модель AS-IS

Для моделирования процессов ИС в нотации IDEF0 AS-IS к функциональному блоку были определены следующие данные:

– Входы:

1) входящие данные – основная и необходимая информация для заполнения документации, включающая в себя данные о преподавательском составе, учащихся, мероприятиях, научных и учебно-методических публикациях и т.д.

– Механизмы:

- 1) специалист кафедры;
- 2) компьютер специалиста;
- 3) принтер.

– Управления:

- 1) Должностная инструкции специалиста кафедры по документационному обеспечению кафедры ПИЕиГД ХТИ - филиала СФУ;
- 2) ФЗ №273 – Федеральный закон "Об образовании в Российской Федерации";

– Выходы:

- 1) документы по кафедре – различная отчетная документация, сдаваемая на хранение или проверку.

Далее проводим декомпозицию функционального блока и разбиваем его на четыре отдельных блока, декомпозиция представлена на рисунке 1.3:

1. Внесение и обработка входящей информации.
2. Формирование документов по кафедре.
3. Сохранение документов и подготовка к печати.
4. Печать документов по кафедре.
5. Внедрение в работу документов по кафедре.

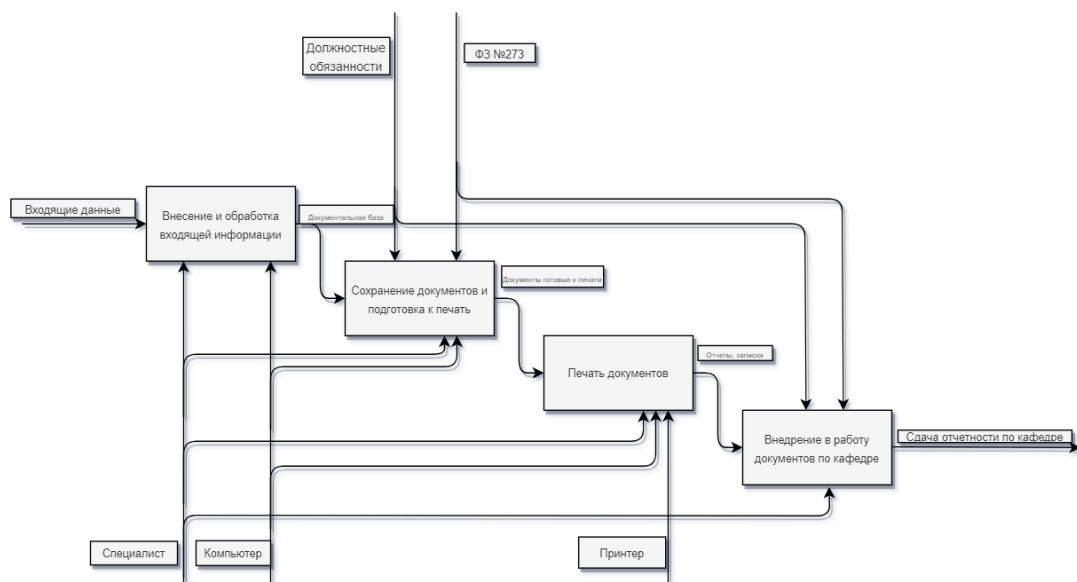


Рисунок 1.3 – Декомпозиция функционального блока «Деятельность специалиста по документационному обеспечению»

К декомпозиции функционального блока были определены следующие данные:

– Входы:

- 1) входящие данные.

– Механизмы:

- 1) Специалист по документационному обеспечению ПИЕиГД.
- 2) компьютер специалиста.
- 3) принтер.

– Управления:

- 1) Должностные инструкции.
- 2) ФЗ №273.

– Выходы:

- 1) сдача отчетов по кафедре.

– Функциональные блоки:

- 1) внесение и обработка входящей информации.

- 2) формирование документов по кафедре.
- 3) сохранение документов и подготовка их к печати.
- 4) печать документов по кафедре.
- 5) внедрение в работу документов по кафедре.

Специалист по документационному обеспечению составляет документы для отчетности в разных квартальных периодах. Зная необходимые данные о мероприятиях, работах преподавателей и студентов он составляет необходимую отчетную документацию, в которой содержится информация о научных работах и научных мероприятиях.

1.6 Обоснование необходимости автоматизации рабочего места специалиста по документационному обеспечению кафедры

В результате анализа деятельности кафедры, его функциональной структуры как объекта автоматизации рабочего места, а также информационной модели работы специалиста по документационному обеспечению были выявлены следующие недостатки:

1. Большие затраты времени и внимания специалиста кафедры на обработку информации и заполнение необходимых документов.
2. Отсутствие базы данных по преподавателям, студентам и документам.
3. Слабое и неэффективное использование технических средств (компьютера) для выполнения поставленных задач.

Работа специалиста по документационному обеспечению кафедры представляет из себя монотонную работу с большим объемом печатной информации, это может повлиять на скорость заполнения документации и вероятных ошибок в тексте, если учитывать человеческий фактор. Из-за этого всегда появляется причина для перепроверки заполненной информации на ее достоверность. Эта часть занимает немалую часть рабочего времени и сил специалиста, что сказывается на его работе.

Кроме того, информация о научной деятельности профессорско-преподавательского состава кафедры поступает не регулярно, а по мере ее появления. В связи с этим документы и сведения хранятся в разрозненном виде. При необходимости их объединения в единый отчет возникает проблема поиска данных.

Смотря на минусы выше, уже вырисовывается необходимость в автоматизации рабочего места специалиста по документационному обеспечению кафедры, что даст нам возможность хранить, обрабатывать и использовать информацию в заполнении документов и также повысить достоверность данных, минимизируя человеческий фактор и убирая большую часть трудоемкости со специалиста, что повысит качество документов для профессорско-преподавательского состава.

1.7 Требования к АРМ специалиста по документационному обеспечению кафедры. Постановка цели и задач

В системе должно быть реализовано следующее:

- 1) внесение в базу данных информации и работа с ней;
- 2) заполнение шаблонов документов для их хранения и печати.

Для правильного создания АРМ специалиста по документационному обеспечению кафедры необходимо использовать следующие нормативные документы, регламентирующие этот процесс:

- Нотация описания бизнес-процессов IDEF0.
- ISO / IEC 12207: 1995-08-01.
- ГОСТ Р ИСО/МЭК 12207-99 «Информационная технология. Процессы жизненного цикла программных средств».

Перед началом разработки разработчику нужно уточнить цели и задачи разрабатываемого продукта. Для создания АРМа специалиста по

документационному обеспечению кафедры ПИЕиГД были уточнены и сформированы следующие цели и задачи.

Цель: улучшение качества работы специалиста по документационному обеспечению кафедры ПИЕиГД ХТИ филиала СФУ путем разработки автоматизированного рабочего места (модуль научно-педагогической деятельности профессорско-преподавательского состава кафедры).

Задачи:

- Проанализировать деятельность ХТИ – филиала СФУ.
- Выполнить сравнительный анализ АРМ, работающих в похожей сфере деятельности. Поставить цели и задачи.
- Выбрать средства разработки.
- Разработать АРМ специалиста кафедры ХТИ – филиала СФУ.
- Рассчитать затраты и экономическую выгоду проекта, определить риски, которым подвержен проект.

1.8 Сравнительный анализ АРМ, работающих в похожей сфере деятельности

После тщательного анализа и рассмотрения популярных программных продуктов для управления документооборотом, таких как 1С:Документооборот 8 ПРОФ, LanDocs и OPTIMA-WorkFlow, стало очевидно, что они не соответствуют нашим требованиям и оказываются слишком дорогими для заказчика. Несмотря на то, что эти решения предлагают широкий спектр функциональности, многие из этих функций являются излишними для нашего использования, а некоторых функций попросту не хватает. Кроме того, их цены находятся в значительном диапазоне - от 45000 рублей для 1С:Документооборот и 8 ПРОФ, а также до 270000 рублей для LanDocs.

Было принято решение разработать собственное автоматизированное рабочее место для системы управления документооборотом, которая будет

точно соответствовать нашим потребностям и бюджету. Это позволит нам избежать необходимости продления лицензий и снизить затраты на установку и настройку, в добавок будет полностью удовлетворять нашим потребностям и желаниям.

Разработка собственной АРМ даст нам возможность создать собственную систему для работы с документами, которая будет легко интегрироваться и обеспечивать оптимальную производительность и удобство использования.

Выводы по разделу «Анализ предметной области и деятельности кафедры ПИЕиГД ХТИ – филиала СФУ»

В данном разделе проанализирована основная деятельность Хакасского технического института – филиала федерального государственного автономного образовательного учреждения высшего образования «Сибирский федеральный университет». Кафедра ПИЕиГД в структуре ХТИ – филиала СФУ является одной из самых крупных. В настоящий момент рабочее место специалиста по документационному обеспечению кафедры нуждается в автоматизации, чтобы оптимизировать работу специалиста, сократив время на заполнение документации и создание отчетов.

Завершением данного раздела стало выявление требований к системе, постановка цели и задач.

2 Разработка АРМа специалиста по документационному обеспечению кафедры ПИЕиГД ХТИ – филиала СФУ

2.1 Модель новой организации процесса ТО-ВЕ

Примером синтеза различных функций в разрабатываемом АРМе будет диаграмма IDEF0. Диаграмма изображена на рисунке 2.1.

Для моделирования процессов разрабатываемого АРМа в IDEF0 к функциональному блоку были выявлены следующие данные:

– Вход:

1) входящие данные.

– Механизмы:

1) специалист кафедры;

2) система АРМ;

3) принтер.

– Управления:

1) федеральный закон №273;

2) должностные инструкции;

– Выход:

1) результаты научной деятельности профессорско-преподавательского состава кафедры;

2) отчеты о результатах научной деятельности профессорско-преподавательского состава кафедры.



Рисунок 2.1 – IDEF0 TO-VE диаграмма разрабатываемого АРМа

Далее проводим декомпозицию функционального блока и разбиваем его на шесть отдельных блоков (рис. 2.2):

- 1) внесение материалов по научной деятельности и их обработка;
- 2) формирование документов;
- 3) сохранение документов и подготовка их к печати;
- 4) печать документов по научной деятельности;
- 5) внедрение в работу документов по научной деятельности.

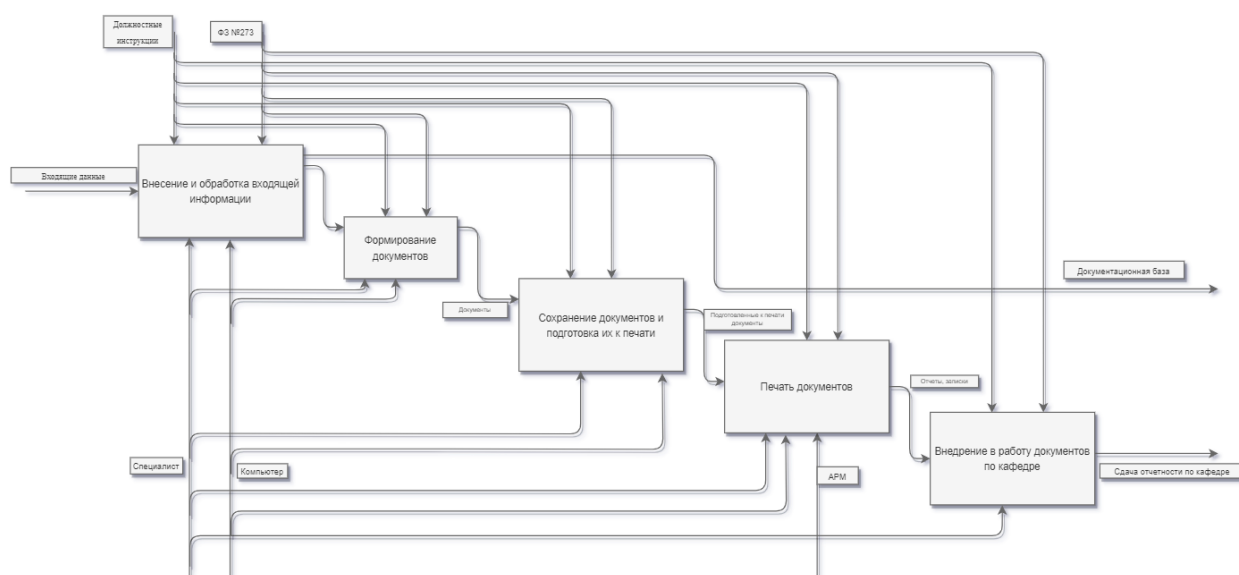


Рисунок 2.2 – Декомпозиция функционального блока ТО-ВЕ

К декомпозиции функционального блока были выявлены следующие данные:

– Вход:

- 1) входящие данные.

– Механизмы:

- 1) специалист кафедры.
- 2) система АРМ.
- 3) принтер.

– Управления:

- 1) федеральный закон №273.
- 2) должностные инструкции.

– Выход:

- 1) результаты научной деятельности.
- 2) отчеты о результатах научной деятельности.

Сначала специалист по документационному обеспечению работает с заполнением и обработкой сведений о научной деятельности профессорско-преподавательского состава кафедры, после чего переходит к формированию документов. Далее после формирования документа он переходит к подготовке к печати.

После того как специалист подготовил документы к печати, он их может распечатать через MicrosoftWord, после чего специалисту остается лишь внедрить распечатанные документы в образовательную деятельность.

2.2 Диаграмма потоков данных

Диаграммы потоков данных показывают, как каждый процесс со временем меняет свои входные данные в выходные, и ставят связующие причины. То есть здесь описывается в основном не процесс, а непосредственно сам поток данных. Диаграмма потоков данных описывает любые действия, в том числе:

- 1) процесс работы с БД;
- 2) работу с информацией, поступающей от преподавателей кафедры.

Нотация DFD состоит из:

- 1) процесса – функция или последовательность действий для обработки данных;
- 2) внешних сущностей – не входящие объекты в систему, но являющиеся для самой системы источниками информации;
- 3) хранилищ данных;

4) потоков данных – стрелками направлений показывает входящую и исходящую информацию.

На рисунке 2.3 построена контекстная диаграмма. Контекстная диаграмма потоков данных содержит процесс автоматизации рабочего места специалиста по документационному обеспечению кафедры ПИЕиГД ХТИ – филиала СФУ.

Внешние сущности:

- 1) специалист кафедры.



Рисунок 2.3 – DFD диаграмма разрабатываемой АРМ

Реестр – информационный ресурс (БД), включающий в себя документы на бумажных и электронных носителях.

Декомпозиция DFD диаграммы разрабатываемой системы представлена на рисунке 2.4.

В декомпозицию входят следующие процессы:

- 1) формирование документов.

Внешние сущности:

- 1) специалист кафедры.

Накопители данных:

1) база данных;

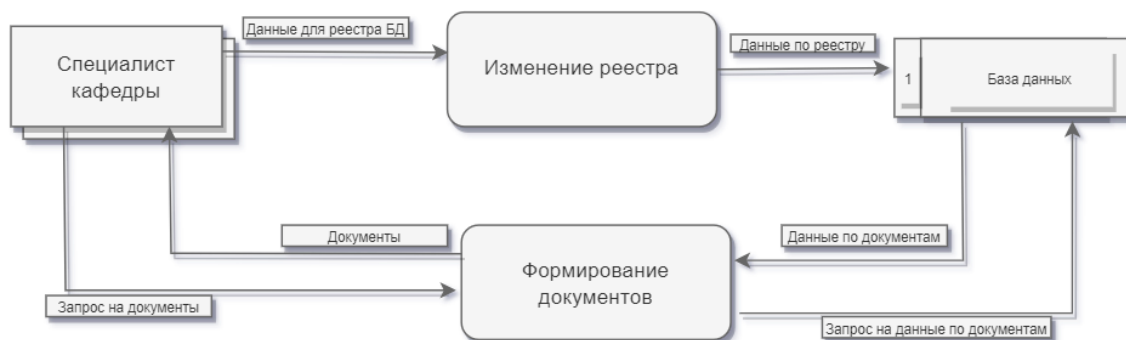


Рисунок 2.4 – DFD декомпозиция диаграммы разрабатываемого АРМа

2.3 Проектирование базы данных. Разработка модели «сущность-связь»

Основными понятиями ER-модели являются сущность, связь и атрибут. Сущность – объект в котором хранится информация доступная и сохраняемая на неопределенный срок. В диаграммах ER-модели сущность представляется в виде прямоугольника, содержащего имя сущности.

В БД хранится информация о студентах, преподавателях, типах документов. Сущность «Преподаватели» представлена в таблице 2.1.

Таблица 2.1 - Атрибуты сущности «prepods»

АТРИБУТ	ОПИСАНИЕ
id_prepod	Уникальный номер для идентификации
prep_surname	Фамилия
prep_name	Имя
prep_3rdname	Отчество
prep_number	Номер телефона
prep_mail	Адрес электронной почты
prep_title	Должность
prep_rank	Ученое звание
prep_step	Ученая степень

Далее приведена таблица 2.2 с сущностью студентов.

Таблица 2.2 – Атрибуты сущности «students»

АТРИБУТ	ОПИСАНИЕ
id_student	Уникальный номер для идентификации
stud_surname	Фамилия
stud_name	Имя
stud_3rdname	Отчество
id_group	Учебная группа

Для формирования документов нам нужны определенные данные для каждого типа документа, поэтому создадим несколько сущностей под каждый тип документа. Перечислим все типы, которые нам нужны: «Научные мероприятия», «Олимпиады», «Издания», «Статьи». И справочники к ним, чтобы в выпадающем списке приписывать определенное качество: «Справочник уровня мероприятий», «Справочник уровня олимпиад», «Справочник уровня изданий» и «Справочник вида изданий», «Справочник уровня статей».

Перечислим атрибуты сущности «Научные мероприятия» и справочник, связанные с ней в таблицах 2.3, 2.4.

Таблица 2.3 – Атрибуты сущности «edition»

АТРИБУТ	ОПИСАНИЕ
id_edition	Уникальный номер для идентификации
id_edition_degree	Уникальный номер для уровня издания
exit_data	Выходные данные
public_year	Год публикации
edition_name	Название издания
volume_edited_lists	Объем

Таблица 2.4 – Атрибуты сущности «sprav_edition_degree»

АТРИБУТ	ОПИСАНИЕ
id_edition_degree	Уникальный номер для идентификации
edition_degree_name	Название уровня издания

Далее в таблицах 2.5, 2.6 идет следующий тип документа «Олимпиады» и также, как и в прошлый раз справочник.

Таблица 2.5 – Атрибуты сущности «olympic»

АТРИБУТ	ОПИСАНИЕ
id_olympic	Уникальный номер для идентификации
id_olympic_degree	Уникальный номер для уровня олимпиады
olympic_name	Название олимпиады
olympic_place	Место проведения

Таблица 2.6 – Атрибуты сущности «sprav_olympic_degree»

АТРИБУТ	ОПИСАНИЕ
id_olympic_degree	Уникальный номер для идентификации
olympic_degree_name	Название уровня олимпиады

Далее в таблице 2.7, 2.8 рассмотрим сущность «Статьи» и «Справочник уровня статей».

Таблица 2.7 – Атрибуты сущности «article»

АТРИБУТ	ОПИСАНИЕ
id_article	Уникальный номер для идентификации
id_article_degree	Уникальный номер для уровня статьи
exit_data	Выходные данные
public_year	Год публикации
article_name	Название статьи

Таблица 2.8 – Атрибуты сущности «sprav_article_degree»

АТРИБУТ	ОПИСАНИЕ
id_article_degree	Уникальный номер для идентификации
article_degree_name	Название уровня статьи

Также необходимо определить связи между преподавателями и типами документов, тип связи будет «много-ко-многим», поэтому нужна связующая таблица 2.9 и 2.10.

Таблица 2.9 – Атрибуты сущности «prepods_article»

АТТРИБУТ	ОПИСАНИЕ
id	Уникальный номер для идентификации
id_article	Номер идентификации статьи
id_prepod	Номер идентификации преподавателя

Таблица 2.10 – Атрибуты сущности «prepods_edition»

АТТРИБУТ	ОПИСАНИЕ
id	Уникальный номер для идентификации
id_edition	Номер идентификации издания
id_prepod	Номер идентификации преподавателя

Так-как нам нужно будет указывать для олимпиады студентов, создадим связующую таблицу между ними, преподавателями и олимпиадой в таблице 2.11.

Таблица 2.11 – Атрибуты сущности «students_olympic»

АТТРИБУТ	ОПИСАНИЕ
id	Уникальный номер для идентификации
id_olympic	Номер идентификации олимпиады
id_prepod	Номер идентификации преподавателя
id_student	Номер идентификации студента
olympic_date	Дата проведения олимпиады
results	Результат

На рисунке 2.5 представлена логическая схема данных. Все ключевые атрибуты находятся в голубом поле внутри рамки, изображающей сущность.

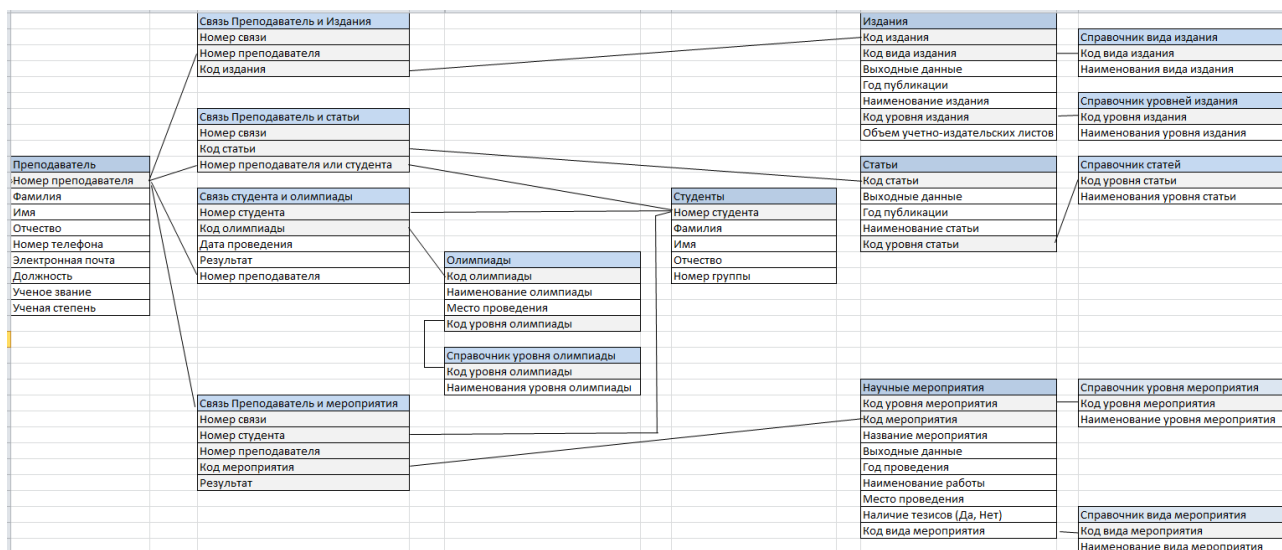


Рисунок 2.5 – Логическая модель данных, построенная на основе методологии ERD

2.4 Выбор программных средств для разработки

Для разработки автоматизированной системы управления (АСУ) был выбран язык программирования Python. Python - один из лучших языков программирования, который обладает рядом преимуществ, таких как высокая скорость выполнения, надежная защита и простота разработки. Он также предоставляет широкий спектр функциональности через свою стандартную библиотеку.

Python поддерживает различные стили программирования, включая структурное, объектно-ориентированное, функциональное программирование. Код на Python организуется в функции и классы, которые могут быть объединены в модули и пакеты, что в свою очередь упрощает читаемость кода.

Он является активно развивающимся. Популярность данного языка подтверждается развитием интерпретатора CPython и его портами для различных платформ, включая Microsoft Windows, UNIX, Mac OS, iOS, Palm OS, OS/2, Amiga, HaikuOS, AS/400, WindowsMobile, Symbian и Android. Там

самым делая из Python отличный вариант для разработки как мобильных, так и компьютерных приложений.

Для разработки на Python мы выбрали интегрированную среду разработки (IDE) PyCharm. PyCharm предоставляет широкие возможности для анализа кода, отладки и запуска юнит-тестов. Она отличается отзывчивым интерфейсом, удобными горячими клавишами и интегрированными функциями по работе с проектами, простотой в установке, малый в габаритах и нагрузках на систему.

Для проектирования базы данных (БД) была выбрана система управления базами данных (СУБД) SQLite. SQLite- это легковесная БД, которая не требует отдельного серверного процесса. Она позволяет использовать язык запросов SQL для доступа к данным.

2.5 Пользовательский интерфейс АРМа

Пользовательский интерфейс будет состоять из шести вкладок приложения:

- 1) Преподаватели.
- 2) Студенты.
- 3) Научные статьи.
- 4) Научные издания.
- 5) Олимпиады.
- 6) Отчеты

Вкладка «Преподаватели» содержит в себе список информации о преподавателях, поля для ввода данных этой информации и кнопки для добавления или редактирования. Вкладка «Преподаватели» представлена на рисунке 2.6.

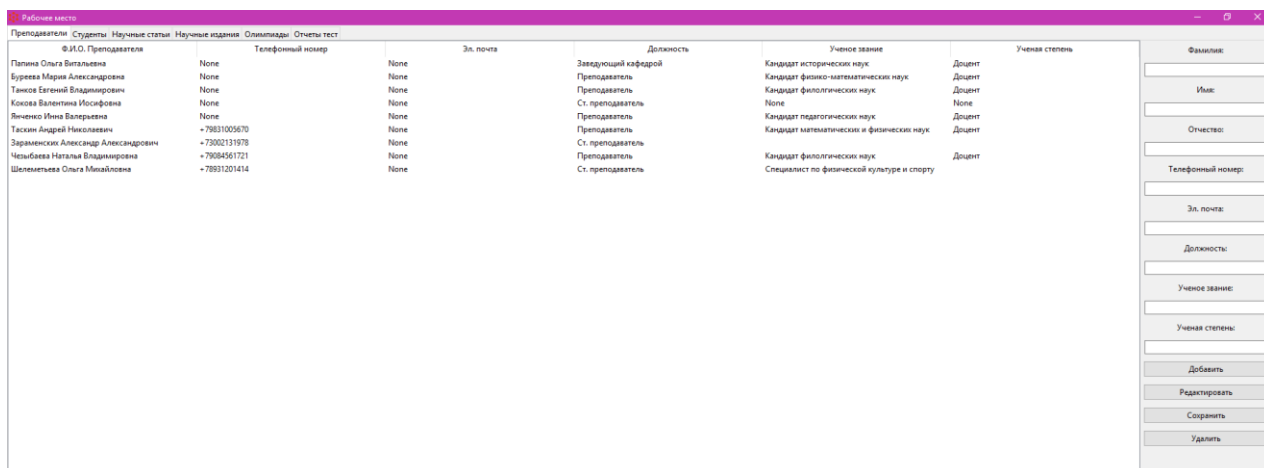


Рисунок 2.6 – Вкладка «Преподаватели»

Вкладка «Студенты» содержит в себе информацию о студентах, их группах, поля для ввода данных этой информации и кнопки для добавления или редактирования. Вкладка «Студенты» представлена на рисунке 2.7.

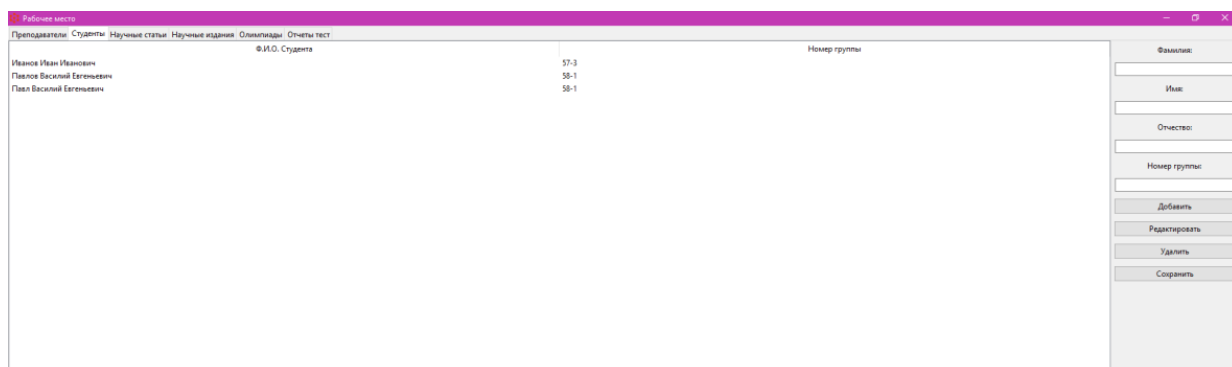


Рисунок 2.7 – Вкладка «Студенты»

Вкладка «Научные статьи» содержит в себе информацию о статьях, поля для ввода информации о них, поля выборки уровня и преподавателей, а также кнопки для создания или редактирования. Вкладка «Научные статьи» представлена на рисунке 2.8.

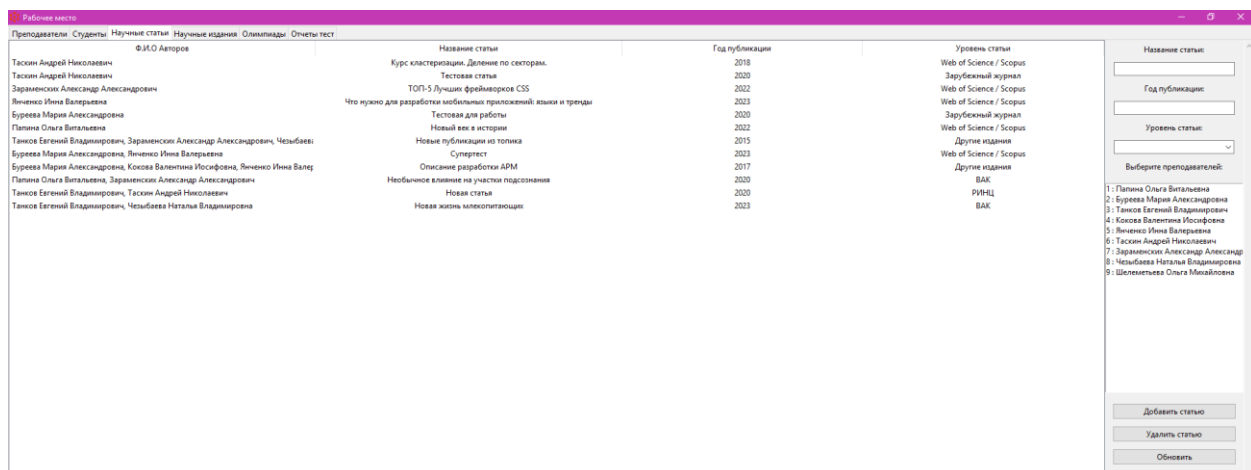


Рисунок 2.8 – Вкладка «Научные статьи»

Вкладка «Издания» содержит в себе информацию об учебно-методических изданиях, поля для ввода информации о них, поля выборки уровня и преподавателей, а также кнопки для создания или редактирования. Вкладка «Научные издания» представлена на рисунке 2.9.

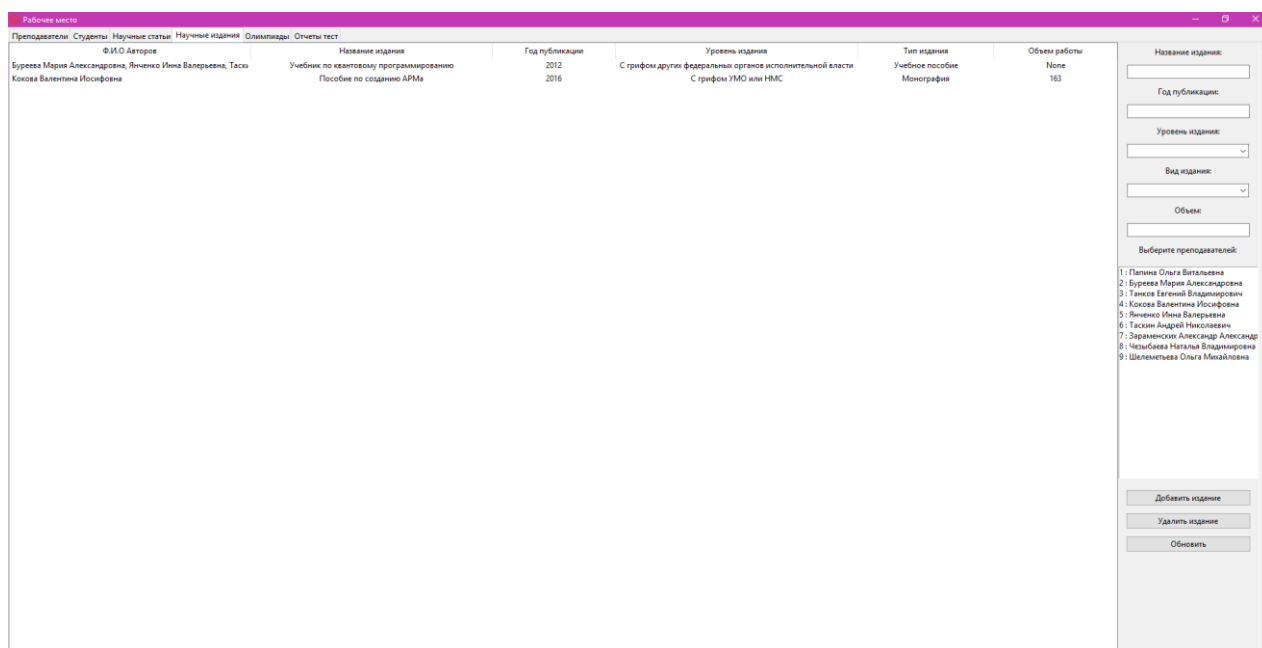


Рисунок 2.9 – Вкладка «Научные издания»

Вкладка «Олимпиады» содержит в себе информацию о добавленных олимпиадах, поля для ввода информации, выборку из студентов и

преподавателей, селектор выбора уровня олимпиады, а также кнопки для добавления и редактирования. Вкладка «Олимпиады» представлена на рисунке 2.10.

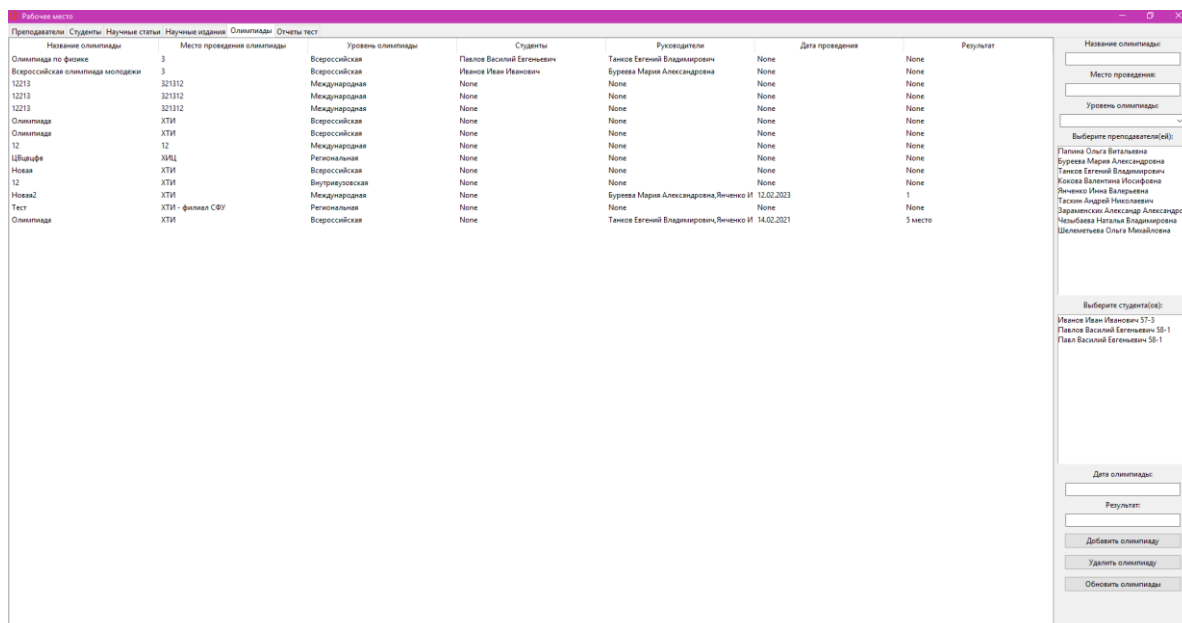


Рисунок 2.10 – Вкладка «Научные издания»

Вкладка «Отчеты тест» содержит в себе 4 кнопки, каждая из которых открывает окно редактирования выбранного типа документа, в которой пользователь может выбрать данные для их экспорта. Вкладка «Отчеты тест» представлена на рисунке 2.11.

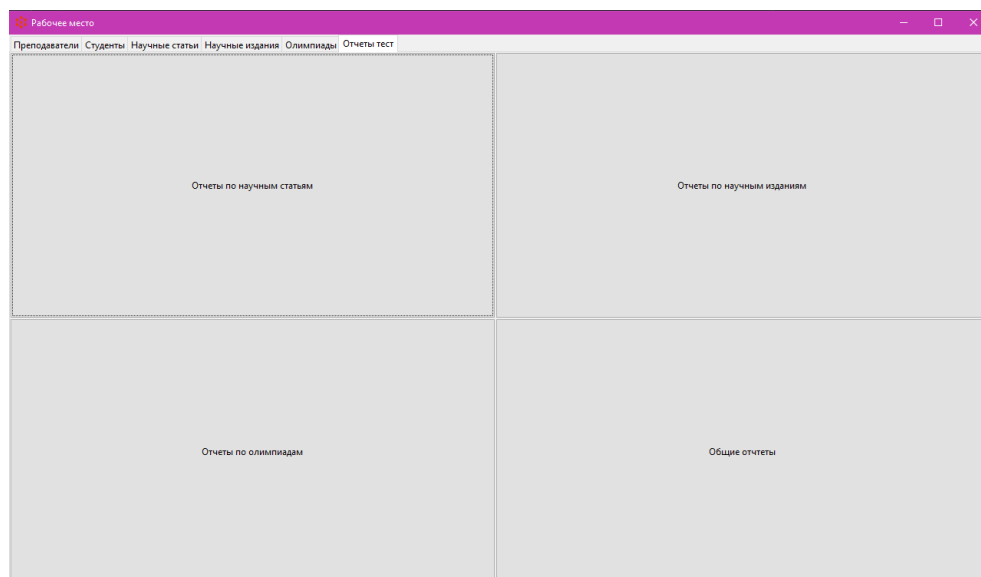


Рисунок 2.11 – Вкладка «Отчеты тест»

2.6 Описание интерфейса АРМа и программных модулей

Далее для объяснения кода будет использоваться символ #, чтобы код можно было удобнее комментировать прямо внутри него.

Для работы с базой данных был написан отдельный модуль, использующий библиотеку sqlite3 (листинг 1):

Листинг 1

```
# Импортирование самого модуля
import sqlite3

# Создание класса для обращения к базе данных
class ARMDDataBase:
    def __init__(self, db='armdb.db'):
        self.conn = sqlite3.connect(db)
        self.cursor = self.conn.cursor()

# Метод для запроса в базу данных, данный метод также возвращает ответы,
# если необходимо получить данные из базы данных
def query(self, _sql):
```

```
self.cursor.execute(_sql)
self.conn.commit()
    _query = self.cursor.fetchall()
    return _query
# Метод для закрытия соединения с базой данных
def close(self):
self.conn.close()
```

2.6.1 Работа с БД

Пример работы с добавлением информации о преподавателях в базу данных. Перейдя на вкладку «Преподаватели», заполним поля справа (рисунок 2.12).

Фамилия:	<input type="text" value="Флеменко"/>
Имя:	<input type="text" value="Остап"/>
Отчество:	<input type="text" value="Николаевич"/>
Телефонный номер:	<input type="text" value="+79080785525"/>
Эл. почта:	<input type="text" value="flameOS@mail.ru"/>
Должность:	<input type="text" value="Преподаватель"/>
Ученое звание:	<input type="text" value="Доцент"/>
Ученая степень:	<input type="text" value="Кандидат физико-математическ"/>
	<input type="button" value="Добавить"/>
	<input type="button" value="Редактировать"/>
	<input type="button" value="Сохранить"/>
	<input type="button" value="Удалить"/>

Рисунок 2.12 - Пример заполнения информации о преподавателе

При нажатии на кнопку добавить мы внесем преподавателя, а также информацию о нем в БД, где сможем использовать эту информацию для других документов. Ниже располагается пример кода для добавления информации (листинг 2):

Листинг 2

#Добавляем функцию, в которой будем брать значения из полей ввода

```
def add_data(self):
```

```
    surname = self.surname_entry.get()
```

```
    name = self.name_entry.get()
```

```
    thirdname = self.thirdname_entry.get()
```

```
    phone = self.phone_entry.get()
```

```
    email = self.email_entry.get()
```

```
    title = self.title_entry.get()
```

```
    rank = self.rank_entry.get()
```

```
    step = self.step_entry.get()
```

```
    #Задаем параметр, что если все элементы заполнены,  
    #то мы добавляем значения в нужную таблицу
```

```
    if surname and name and thirdname and phone and email and title and  
rank and step:
```

```
        self.cursor.execute(
```

```
            "INSERT INTO preps (prep_surname, prep_name,  
prep_3rdname, prep_number, prep_mail, prep_title, prep_rank, prep_step) VALUES  
(?, ?, ?, ?, ?, ?, ?, ?)",
```

```
            (surname, name, thirdname, phone, email, title, rank, step))
```

```
self.conn.commit()
```

```
#Выводим значение ФИО для таблицы в единичное значение
```

```
full_name = f"{surname} {name} {thirdname}"
```

```

        self.treeview.insert("", tk.END, values=(full_name, phone, email,
title, rank, step))
#Очищаемполяввода
        self.surname_entry.delete(0, tk.END)
        self.name_entry.delete(0, tk.END)
        self.thirdname_entry.delete(0, tk.END)
        self.phone_entry.delete(0, tk.END)
        self.email_entry.delete(0, tk.END)
        self.title_entry.delete(0, tk.END)
        self.rank_entry.delete(0, tk.END)
        self.step_entry.delete(0, tk.END)
    else:
        messagebox.showwarning("Ошибка", "Пожалуйста,
заполнитевсеполя.")

```

Данные, в свою очередь, можно редактировать и удалять из базы данных. Разберем пример с удалением информации, для этого все также понадобится tkinter (листинг 3)

Листинг 3

```

#Создаемфункцию
defdelete_data(self):
    selected_item = self.treeview.selection()
#Выбираемданныеизtreeview
    if selected_item:
        values = self.treeview.item(selected_item)["values"]
        iflen(values) == 6:
            result = messagebox.askyesno("Удаление",
"Выдействительнохотитеудалитьэтузапись?")

```

```

ifresult:
#ОперациясБД, вкотороймыудаляемзаписьизтаблицыпоФИОпреподавателя
    self.cursor.execute(
        "DELETEFROMprepodsWHEREprep_surname=? AND prep_name=?
AND prep_3rdname=?",
        (values[0], values[1], values[2]))
    self.conn.commit()
    self.treeview.delete(selected_item)
    messagebox.showinfo("Удаление", "Записьуспешноудалена.")
else:
    messagebox.showwarning("Ошибка", "Пожалуйста, выберите запись
для удаления.")

```

Пример работы с научными статьями. Перейдя во вкладку «Научные статьи» и выбрав селектор уровня статьи, он подгрузит нам данные об уровне статьи, которые хранятся в справочнике базы данных. Также, в этой же вкладке представлен селектор выбора преподавателя, данные которого мы берем из созданных нами и сохраненных в БД и привязываем созданную статью.

Пример использования БД во вкладке «Научные статьи» представлен на рисунке 2.13.

Рисунок 2.13 – Интерфейс селекторов с БД

Чтобы селектор преподавателей подгрузился, использовался следующий код (листинг 4):

Листинг 4

Задаем переменную, в которой будет содержаться запрос на выбор уровня статьи из справочника

```
defget_article_degrees(self):
```

```
#Подключаемсякбазеданных
```

```
conn = sqlite3.connect('armdb.db')
```

```
cursor = conn.cursor()
```

```

#Создаем запрос на выбор id уровня и его названия
select_query = "SELECT id_article_degree, article_degree_name FROM
sprav_article"
cursor.execute(select_query)
article_degrees = cursor.fetchall()

conn.close()

#Задаем формат листа, в котором при раскрытии списка будет показываться
первый элемент таблицы (id)и второй (название)
article_degrees_list = [{"{} : {}".format(degree[0], degree[1]) for degree in
article_degrees]
returnarticle_degrees_list

#Создаемфункциюcombobox, избиблиотекиTkinter, который работает как
выпадающий список и ставим ему значение «readonly» благодаря которому мы
можем только выбирать значения из списка, а не заносить их.
self.combobox_article_degree = ttk.Combobox(self, state="readonly", width=27)
self.combobox_article_degree.pack(pady=5)

#В завершении создадим метод для обновления значения Combobox, после того
как выполним команду
defupdate_combobox(self):
self.combobox_article_degree['values'] = self.get_article_degrees()
self.listbox_prepod_select.delete(0, tk.END)
self.listbox_prepod_select.insert(tk.END, *self.get_prepods_list())

```

В этой же вкладке создадим функцию Listbox для преподавателей, т.к. это позволит нам делать мультिवыбор переменных для заполнения информации в БД (листинг 5).

Листинг 5

```

#Создаем сам Listbox и указываем в нем множественную выборку
self.listbox_prepod_select = tk.Listbox(self, height=20, width=35,
selectmode=tk.MULTIPLE)
self.listbox_prepod_select.pack(fill=tk.BOTH, pady=10)
#По аналогии с предыдущей выборкой, напишем такую же, только для
преподавателей
def get_prepods_list(self):
    conn = sqlite3.connect('armdb.db')
    cursor = conn.cursor()
    #Запрос на выбор из БД
    select_query = "SELECT id_prepod, prep_surname, prep_name, prep_3rdname
FROM prepods"
    cursor.execute(select_query)
    prepods = cursor.fetchall()
    conn.close()
    #Форматзаписи
    prepods_list = [{"id": prepod[0], "surname": prepod[1], "name": prepod[2], "3rdname": prepod[3]}
for prepod in prepods]
return prepods_list

```

Подобных запросов для отображения в графическом интерфейсе приложения получилось около 15 штук.

2.6.2 Метод для создания элементов в списке

Для того чтобы показать пользователю какие есть статьи, уровни, года публикации и авторы, воспользуемся функцией Treeview, для создания графических таблиц с использованием все той же библиотеки Tkinter (листинг б).

Листинг 6

```
#Создаемметод
defupdate_treeview(self):
self.treeview.delete(*self.treeview.get_children())
#ПодключаемсякБД
    conn = sqlite3.connect('armdb.db')
cursor = conn.cursor()
#Берем множественную выборку значений из БД
select_query = """
    SELECT    article.id_article,    article.article_name,    article.public_year,
sprav_article.article_degree_name,
#Используем group_concatдля объединения ФИО в одно значение
GROUP_CONCAT(prepos.prep_surname || ' ' || prepos.prep_name || ' ' ||
prepos.prep_3rdname, ', ') AS prepod_names
    FROM article
    JOIN prepos_article ON article.id_article = prepos_article.id_article
    JOIN prepos ON prepos_article.id_prepod = prepos.id_prepod
    LEFT JOIN    sprav_article    ON    article.id_article_degree    =
sprav_article.id_article_degree
    GROUP BY    article.id_article,    article.article_name,    article.public_year,
sprav_article.article_degree_name
    """
cursor.execute(select_query)
rows = cursor.fetchall()
    for row in rows:
article_id, article_name, public_year, article_degree, prepod_names = row
#Поочередно записываем как они должны отображаться в таблице
self.treeview.insert("", 'end', values=(prepod_names, article_name, public_year,
article_degree))
```

conn.close()

На создание табличного интерфейса в приложение подобный метод будет использоваться еще 10 раз.

2.6.3 Редактор документов

Во вкладке «Отчеты» у нас есть 4 кнопки, каждая из которых открывает окно с редактированием и экспортом нужного нам типа документа. Для примера рассмотрим вкладку отчетов по научным статьям(рисунок 2.14).

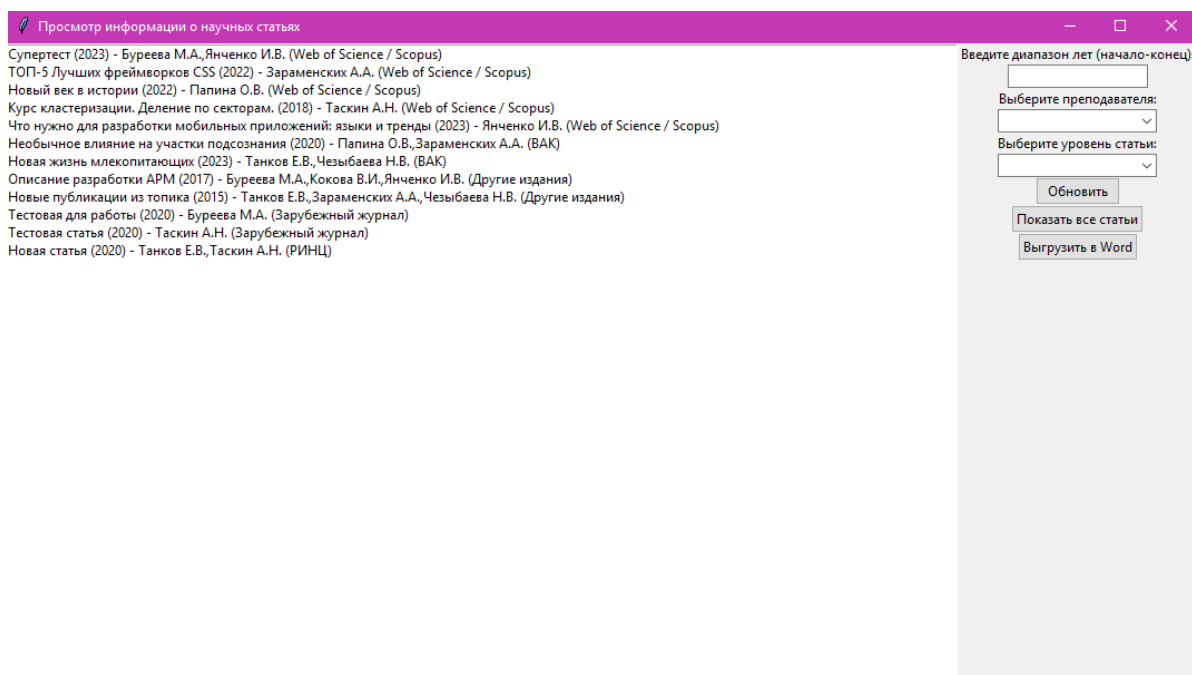


Рисунок 2.14 – Интерфейс окна «Отчеты по статьям»

Поля для ввода значений нужны для сортировки статей в базе данных. «Введите диапазон лет (начало-конец)» сортирует значения по дате публикации в формате «год-год». «Выберите преподавателя» показывает статьи, которые привязаны к авторству в выпадающем списке «Combobox», такой же метод применяется для выбора уровня статьи. Кнопка «Обновить» обновляет

значения листа слева и применяет выбранные фильтры, «Показать все статьи» полностью обновляет лист, сбрасывая введенные фильтры. При нажатии «Выгрузить в Word» программа выгружает выходные данные статей в уже готовый шаблон документа.

Для примера посмотрим код для выгрузки документа как таблица, используя библиотеку python-docx (листинг 7).

Листинг 7

```
def export_to_word(self):
    #Проверяем информацию о выбранной статье
    if not self.selected_article_info:
        messagebox.showerror("Ошибка", "Не удалось получить информацию о статье.")
    return
    article_info = self.selected_article_info
    #Создаем новый документ типа .docx
    document = Document()
    #Делаем таблицу, используя функционал python-docx
        document.add_heading('Информация о статье', 0)
            table = document.add_table(rows=2, cols=2)
            table.style = 'Table Grid'
            cell1 = table.cell(0, 0)
            cell1.text = 'Название статьи:'
            cell2 = table.cell(0, 1)
            cell2.text = article_info['article_name']
            cell3 = table.cell(1, 0)
            cell3.text = 'Год публикации:'
            cell4 = table.cell(1, 1)
            cell4.text = article_info['article_year']
    document.add_paragraph('\n')
```

```
document.add_heading('Авторыстатьи', level=1)
    document.add_paragraph(article_info['prepods_names'])
    document.add_paragraph('\n')
    document.add_heading('Уровеньстатьи', level=1)
document.add_paragraph(article_info['article_degree'])
#Загружаем значения в уже готовый документ, если нет то создается
«article.docx»
document.save('article.docx')
messagebox.showinfo("Успех", "Статья успешно выгружена в Word.")
```

Пример статей, экспортированных в .docx файлы (рисунок 2.15, рисунок 2.16).

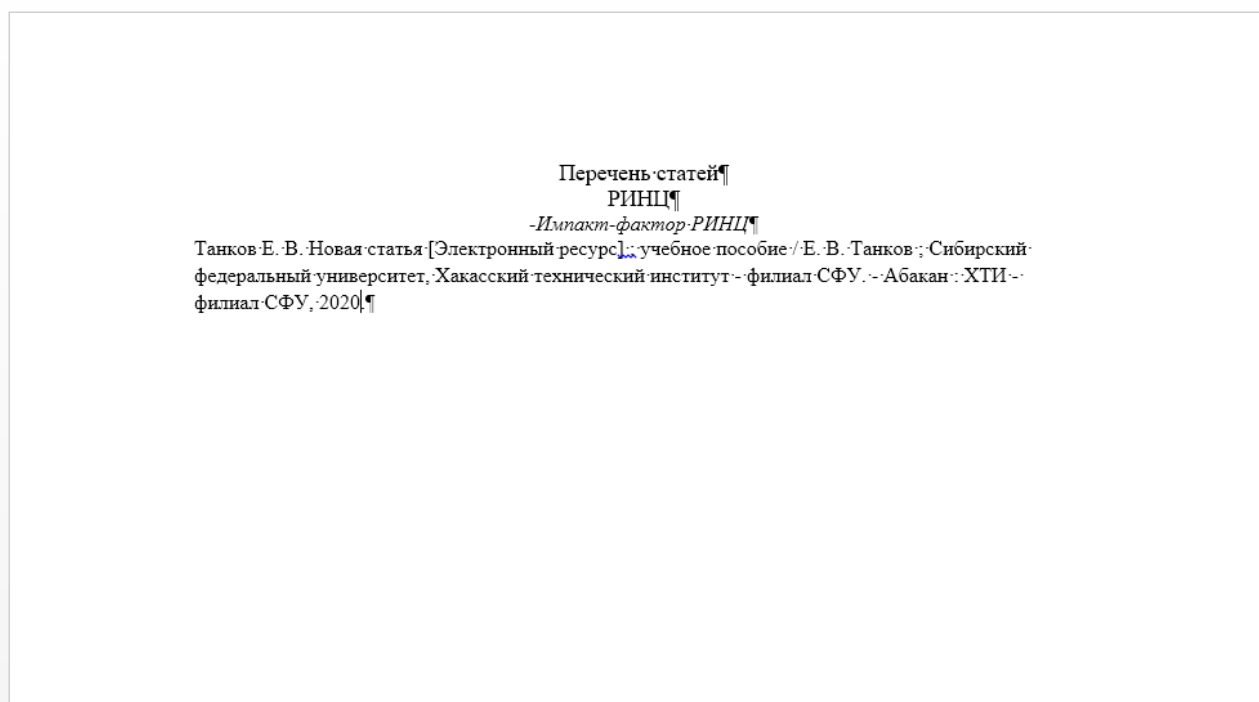


Рисунок 2.15 – Перечень статей РИНЦ

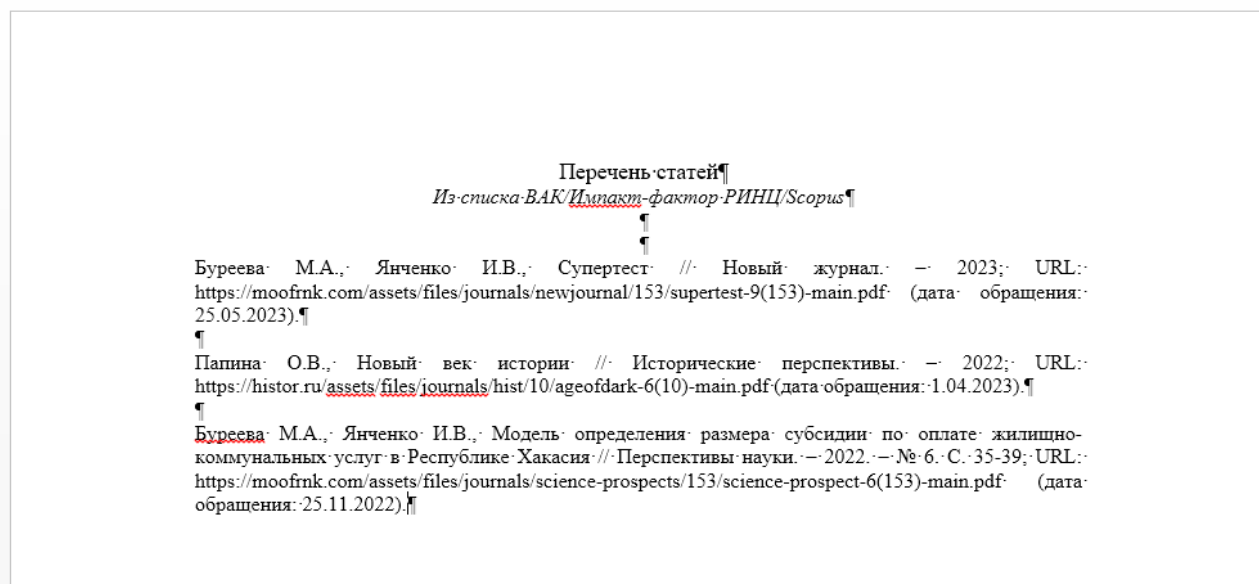


Рисунок 2.16 – Перечень статей ВАК

Выводы по разделу «Разработка АРМа специалиста по документационному обеспечению кафедры ПИЕиГД ХТИ – филиала СФУ»

В этом разделе разработан и описан АРМ специалиста по документационному обеспечению кафедры ПИЕиГД ХТИ – филиала СФУ.

В результате были выполнены следующие работы:

1. Спроектирован и разработан интерфейс АРМа, содержащий около 10 различных модулей для работы с информацией о научной деятельности профессорско-преподавательского состава кафедры.

2. Упрощена работа с БД путем создания среды работы в ней.

3. Написаны функции и методы для работы с интерфейсом.

4. Написаны функции и методы для документирования.

Для реализации АРМа специалиста кафедры были проанализированы различные среды программирования

Для проектирования БД выбрана СУБД SQLite, так как она является бесплатной и имеет пользовательский интерфейс, для работы с ней.

Для разработки АРМ был выбран язык Python, т.к. он очень компактный, простой и удобный в разработке за счет уже готовых библиотек и функций.

Для разработки на языке Python был выбран IDE PyCharm, он обладает легким рабочим интерфейсом, простотой загрузки библиотек, к тому же является бесплатным.

3 Расчет экономических затрат на реализацию АРМа специалиста по документационному обеспечению кафедры ПИЕиГД ХТИ – филиала СФУ

Для разработки АРМа специалиста по документационному обеспечению кафедры ПИЕиГД ХТИ – филиала СФУ очень важной является проблема оценки материальных затрат на успешное завершение проекта. Существует огромное количество разнообразных методик для выполнения оценки стоимости, из этого множества можно выделить традиционную оценку затрат и экономической эффективности [10].

Сроки реализации проекта:

- консультация с заказчиком, анализ предметной области – 4 дня;
- разработка прототипа – 20 дней;
- доработка – 4 дней;
- тестирование – 2 дня;
- завершение и введение в эксплуатацию – 5 дней.

3.1 Методика оценки совокупной стоимости разработки АРМа

Для расчета технических и экономических затрат на разработку была выбрана методика TCO (Total Cost of Ownership – рус. совокупная стоимость владения или рус. стоимость жизненного цикла). Ключевым принципом, реализуемым при разработке методик определения совокупной стоимости владения, является системный подход. TCO рассчитываются по формуле

$$TCO = DE + IC_1 + IC_2, \quad (1)$$

где DE (direct expenses) – прямые расходы;
IC₁, IC₂ (indirect costs) – косвенные расходы (1 и 2 группы).

3.2 Капитальные затраты

Капитальные затраты – единовременные затраты.

Капитальные затраты рассчитываются по формуле

$$K = K_{\text{пр}} + K_{\text{тс}} + K_{\text{лс}} + K_{\text{по}} + K_{\text{ио}} + K_{\text{об}} + K_{\text{оэ}}, \quad (2)$$

где $K_{\text{пр}}$ – затраты на проектирование ПО;
 $K_{\text{тс}}$ – затраты на технические средства управления;
 $K_{\text{лс}}$ – затраты на создание линий связи, а также интернет-соединения;
 $K_{\text{по}}$ – затраты на программные средства для использования готового программного продукта;
 $K_{\text{ио}}$ – затраты на формирование информационной базы;
 $K_{\text{об}}$ – затраты на обучение персонала;
 $K_{\text{оэ}}$ – затраты на опытную эксплуатацию.

$K_{\text{тс}} = 0$ рублей. Не требуется поскольку заработная плата специалиста кафедры и затраты на ПО по управлению проектом не входят в расчет денежных средств заказчика.

$K_{\text{лс}} = 0$ рублей. Не требуется, так как затрат на создание локальных сетей и интернет-соединений не требуется.

$K_{\text{по}} = 0$ рублей. Не требуется, поскольку будет использоваться предустановленное ПО на компьютере специалиста.

$K_{\text{ио}} = 0$ рублей. Информация будет вноситься в систему АРМ в процессе эксплуатации.

Затраты на проектирование программного обеспечения рассчитываются по формуле

$$K_{\text{пр}} = K_{\text{зп}} + K_{\text{ипс}} + K_{\text{свт}} + K_{\text{проч}}, \quad (3)$$

где $K_{\text{зп}}$ – затраты на заработную плату разработчика;

$K_{\text{ипс}}$ – затраты на инструментальные, программные средства для проектирования;

$K_{\text{свт}}$ – затраты на средства вычислительной техники для проектирования;

$K_{\text{проч}}$ – прочие затраты на проектирование и разработку. Возьмем примерно 5% к стоимости, в расчет на электроэнергию и коммуналку.

Чтобы спроектировать и разработать АРМ специалиста по документационному обеспечению кафедры ПИЕиГД ХТИ – филиала СФУ понадобятся услуги программиста, который пользуется языком программирования Python, также в идеале чтобы он пользовался средой программирования (IDE) PyCharm.

1) разработка интерфейса и функционала АРМ с учетом требований заказчика и составленного технического задания;

2) разработка и воплощение тестов;

У разработчика на работы будет 1 месяц, за который он должен будет выполнить свои задачи.

Для расчета заработной платы разработчика необходимо будет составить таблицу заработной платы.

ФОТ (Фонд оплаты труда) составляет сумму выплат заработной платы разработчика и 30,2% от этой суммы – обязательные платежи во внебюджетные фонды.

При расчете затрат на оплату труда разработчика мы должны увеличить величину оклада 16242 рублей на 60%, так как разработка ведется в Республике Хакасия, где районный коэффициент равен 30%, по такому же принципу действует северный коэффициент, который равен 30%. Не забудем учесть отчисления во внебюджетные фонды, которые составляют 30,2%. Рабочая норма одного разработчика в месяц 22 дня [1, 13].

На основании расчета заработной платы одного разработчика получаем:

$$K_{\text{зн}} = 25987 * 1,302 = 33835 \text{ рублей,}$$

Таблица 3.1 – Расчет заработной платы разработчика

Состав заработной платы	Сумма, руб.
Оклад	16242
Районный коэффициент	4873
Северный коэффициент	4873
НДФЛ (13%)	3378
Итого	25987

Также для разработки понадобится внутренне ПО для работы разработчика с кодом.

Расчет затрат на программное и инструментальное обеспечение указаны в таблице 3.2.

Таблица 3.2 – Стоимость программного обеспечения

Наименование программного обеспечения	Стоимость, с учетом периода использования	Период использования
PyCharm	Free, OS	Без срока использования
PyInstaller	Free (Стандартный установщик)	Без срока использования
Python	Free (Python Software Foundation License)	Без срока использования
Tkinter	Free (GPL v3)	Без срока использования
Python-docx	Free (GPL v3)	Без срока использования
SQLiteStudio	Free, OS	Без срока использования

Получается, что $K_{\text{инс}} = 0$ рублей.

Расчет затрат на средства вычислительной техники ($K_{\text{свт}}$) выполним, опираясь на минимальные системные требования на компоненты из таблицы 3.3.

Таблица 3.3 – Системные требования программных средств

Наименование программного обеспечения	PyCharm	Microsoft Windows 10 Home x64
Рекомендуемые системные требования	64-битная версия Microsoft Windows 10, 8, 7.	Процессор 1,2Гц.
	Оперативная память 6 ГБ.	Оперативная память 2 ГБ.
	2,5 ГБ. свободного места на диске	32 ГБ места на жестком диске
	Разрешение экрана 1024x768 пикселей	Разрешение экрана 800x600 пикселей
	Python 3.5+	Видекодек Direct 9+

По данным из таблицы выше, составим таблицу 3.4, в которой укажем полную стоимость ПО

Таблица 3.4 – Стоимость аппаратного обеспечения

Номер оборудования	Наименование элементов оборудования	Количество элементов, шт	Стоимость единицы, руб	Срок службы, лет
3.4.1	Готовый системный процессор	1	15912	5
3.4.2	Монитор + HDMI кабель	1	8199	5
3.4.3	Клавиатура Redragon ES-0	1	1399	1
3.4.4	Мышь проводная Defender	1	450	1
Итого		1	25960	

В проекте стоимость оборудования будет рассчитываться через амортизационные отчисления.

Рассчитаем ставку амортизации на каждый элемент оборудования отдельно по формуле

$$H_a = (1 / \text{срок службы}) * 100\%. \quad (4)$$

Далее рассчитываем годовую амортизацию

$$A_T = \text{себестоимость} * N_a. \quad (5)$$

Теперь рассчитаем общий объем амортизации по формуле

$$A_{\text{общ}} = (A_{T1} + A_{T2} + \dots + A_{TN}) / 12 * \text{срок работ} \quad (6)$$

В таблице 3.5 проведем расчеты амортизационных отчислений.

Таблица 3.5 – Стоимость программного обеспечения

Номер элемента оборудования	Количество элементов, шт	Стоимость единицы, руб	Срок службы, лет	Ставка по амортизации и по формуле (4)	Годовая амортизация по формуле (5)
3.4.1	1	15912	5	20	3182,4
3.4.2	1	8199	5	20	1639,8
3.4.3	1	1399	1	100	1399
3.4.4	1	450	1	100	450
Итого		25960			

$A_{\text{общ}}$ по формуле (6): 556рублей.

При расчетах в таблице 3.5 получили, что $A_{\text{общ}} = 556$ рублей, выходит, $K_{\text{свт}} = 556$ рублей.

Проведем расчет общих затрат, для этого

$$K_{\text{проч}} = (33835 + 556) * 0,05 = 1720 \text{рублей.}$$

После подсчета всех затрат, проведем расчет, опираясь на формулу (3)

$$K_{\text{пр}} = 33835 + 556 + 1720 = 36111 \text{рублей.}$$

Состав проектных затрат по компонентам изображен на рисунке 3.1.



Рисунок 3.1 – Проектные затраты на реализацию проекта

Исходя из этого получаем то, что на зарплату разработчика приходится почти вся доля затрат.

$K_{об} = 25987/22/2 * 1,302 = 769$ рублей. Стоимость $\frac{1}{2}$ от рабочего дня разработчика для обучения им специалиста кафедры ПИЕиГД.

$K_{ос} = 25987/22/4 * 1,302 = 385$ рублей. Стоимость $\frac{1}{4}$ от рабочего дня разработчика для установки программного обеспечения и запуска его в работу.

Согласно формуле (2), $K = 36111 + 769 + 385 = 37265$ рублей.

3.3 Эксплуатационные затраты

Эксплуатационные затраты – абсолютный показатель издержек, понесенных компанией на поддержание работоспособности. Являются повторяющимися, и они рассчитываются на цикл производства, чаще всего на 1 год и рассчитываются по следующей формуле

$$C = C_{\text{зп}} + C_{\text{ао}} + C_{\text{то}} + C_{\text{лс}} + C_{\text{ни}} + C_{\text{проч}}, \quad (7)$$

где $C_{\text{зп}}$ – зарплата управленческого персонала, работающего с использованной ИС;

$C_{\text{ао}}$ – амортизационные отчисления;

$C_{\text{то}}$ – затраты на техническое обслуживание;

$C_{\text{лс}}$ – затраты, связанные с использованием глобальных сетей;

$C_{\text{ни}}$ – затраты на носители информации;

$C_{\text{проч}}$ – прочие затраты.

Специалист по документационному обеспечению ежемесячно обрабатывает информацию о научной деятельности профессорско-преподавательского состава кафедры. На эту работу он затрачивает примерно 4 часа каждый месяц. Заработная плата специалиста по документационному обеспечению кафедры ПИЕиГД за 1 год составляет:

$$C_{\text{зп}} = 32624 \text{ рублей} / 22 / 8 * 4 * 1,302 * 12 \text{ месяцев} = 11585 \text{ рублей.}$$

$C_{\text{ао}} = 0$ рублей, т.к. кафедрой уже предусмотрен личный компьютер специалиста по документационному обеспечению.

$C_{\text{то}} = 0$ рублей, т.к. заказчик оплачивает все затраты на техническое обслуживание вне зависимости от внедренной системы.

$C_{\text{лс}} = 0$ рублей, специалист будет работать в локальной сети, подключение к глобальной сети бессмысленно.

$C_{\text{ни}} = 0$ рублей, т.к. все данные будут храниться на компьютере специалиста.

Прочие затраты составляют 3% от суммы предыдущих эксплуатационных затрат:

$$C_{\text{проч}} = 11585 * 3\% = 348 \text{рублей.}$$

Проведем расчет эксплуатационных затрат:

$$C = 11585 + 0 + 0 + 348 = 11933 \text{рубля.}$$

Состав эксплуатационных затрат представлен на рисунке 3.2.



Рисунок 3.2 – Эксплуатационные затраты на реализацию

3.4 Расчет затрат реализации проекта. Методика TCO (Total Cost of Ownership)

Прямые расходы рассчитываются по формуле

$$DE = DE1 + DE2 + DE3 + DE4 + DE5 + DE6 + DE7 + DE8, \quad (8)$$

где DE1 – капитальные затраты;

DE2 – расходы на управление ИТ;

DE3 – расходы на техническую поддержку АО и ПО;

DE4 – расходы на разработку прикладного ПО внутренними силами;

DE5 – расходы на аутсорсинг;

DE6 – командировочные расходы;

DE7 – расходы на услуги связи;

DE8 – другие группы расходов.

DE1 = 37265 рублей, т.к. капитальные затраты посчитаны выше;

DE2 = $C_{зп} = 11585$ рублей;

DE3 = $C_{ао} + C_{то} = 0 + 0 = 0$ рублей;

DE4 = 0 рублей, т.к. расходы на разработку программного обеспечения внутренними силами не предусмотрены;

DE5 = 0 рублей, т.к. сдавать реализацию проекта в сторонние студии не понадобится;

DE6 = 0 рублей, т.к. командировочные не потребуются;

DE7 = 0 рублей, т.к. затраты на услуги связи тоже не понадобятся;

Расходы на другие группы составляют 5% от суммы всех прямых расходов и составляют:

$$DE8 = (37265 + 11585) * 5\% = 2443 \text{ рубля.}$$

По формуле (8) DE = 51293 рубля, подробнее в таблице 3.6.

Таблица 3.6 – Прямые расходы

Группа расходов	Планируемая сумма, руб
Капитальные затраты	37265
Расходы на управление ИТ	11585
Расходы на техническую поддержку АО и ПО	0
Расходы на разработку прикладного ПО внутренними силами	0
Расходы на аутсорсинг	0
Командировочные расходы	0
Расходы на услуги связи	0
Другие группы расходов	2443
Итого:	51293

IC₁, IC₂ (IndirectCosts) – т.к. риски на реализацию проекта на первых этапах проектирования, на внедрении и на этапе эксплуатации крайне низкие, то косвенные расходы IC₁ и IC₂ равны нулю.

Согласно формуле (1)

$$TCO = 51293 + 0 + 0 = 51293 \text{ рубля.}$$

3.5 Риски реализации проекта

В каждом ИТ-проекте, почти на каждом этапе разработки и реализации существуют риски. Данный проект не исключение, так что выделим риски:

1) риск соответствия – он имеет низкую вероятность, так как в процессе выполнения каждый этап работы обговаривался с заказчиком. Так или иначе всегда есть небольшой шанс недопонимания, который может возникнуть на любом этапе разработки. Решение: уточнить детали у заказчика на каждом этапе разработки, при появлении каких-либо заминок;

2) Риски, связанные с характеристиками проекта – данный риск имеет средний уровень влияния. Есть вероятность что программист не успеет в срок выполнять поставленную задачу, и разработчику придется доделывать проект за свои деньги, без оплаты труда. При этом вероятность риска средняя, так как

разработчик может быть недостаточно опытным, что может вызвать задержки в работе. Решение: можно нанять еще одного программиста, это снизит данный риск, но необходимо будет провести перерасчет стоимости реализации проекта.

Риски проекта представлены в таблице 3.7.

Таблица 3.7 – Оценка рисков проекта

Группы рисков	Перечень рисков	Уровень влияния	Вероятность риска	Возможность предотвращения или снижения
Риск соответствия	Недопонимание заказчика и разработчика	Средний	Низкая	Более точное описание желаний заказчика
Риски, связанные с характеристиками проекта	Ограниченное время разработки проекта	Средний	Средний	Наем дополнительного программиста

В результате проведенного анализа были выявлены вероятность и уровень влияния рассмотренных рисков. Также были разработаны возможные пути решения данных рисков.

Выводы по разделу «Расчет экономических затрат на реализацию АРМа специалиста по документационному обеспечению кафедры ПИЕиГД ХТИ – филиала СФУ»

При расчёте оценки экономической эффективности были выполнены следующие расчеты:

– Расчёт капитальных (единовременных затрат). АРМ специалиста по документационному обороту кафедры ПИЕиГД ХТИ – филиала СФУ капитальные затраты составили $K = 37265$ рублей.

– Расчёт затрат на разработку проекта автоматизации процессов составил $K_{пр} = 36111$ рублей.

– Расчет эксплуатационных затрат составил $C = 11933$ рубля.

Произведена оценка рисков проекта. На основе перечня рисков и уровня их влияния на проект рекомендованы мероприятия по снижению вероятности риска.

Разрабатываемый программный продукт подразумевается, как «АРМ – автоматизированное рабочее место». Спроектированное автоматизированное рабочее место экономит около 30% рабочего времени. Раньше на работу специалиста кафедры уходило 8 часов, теперь уходит 6 и экономится примерно 25% в год.

Единственным источником дохода является заказчик ХТИ – филиал СФУ.

ЗАКЛЮЧЕНИЕ

В ходе выполнения выпускной квалификационной работы был создан готовый АРМ (модуль обработки информации о научной деятельности профессорско-преподавательского состава кафедры). Были решены следующие задачи:

- Проанализирована деятельность кафедры ПИЕиГД ХТИ – филиала СФУ.
- Выполнен сравнительный анализ АРМ, работающих в похожей сфере деятельности.

Поставлены цели и задачи.

- Выбраны средства разработки.
- Разработан рабочий АРМ.
- Рассчитаны затраты на реализацию проекта, определены риски, которым подвержен проект.

В первом разделе выпускной квалификационной работы была проанализирована предметная область, определены назначение и характеристики АРМ.

Во втором разделе выпускной квалификационной работы была создана общая структура АРМ, общий интерфейс, выбраны и обоснованы средства проектных решений, описан порядок разработки АРМ.

Разработано около 10 модулей интерфейса, написан модуль для облегчения работы с базой данных. Также реализовано документирование путем ввода данных в шаблон документа.

В третьем разделе выпускной квалификационной работы были рассчитаны капитальные и эксплуатационные затраты, определена стоимость по методике ТСО, определены риски проекта, определены доходы от проекта.

Разработанное автоматизированное рабочее место специалиста по документационному обеспечению кафедры, ускоряет и облегчает работу самого

специалиста путем добавления легкой работы с данными, простому интерфейсу, вкладкам с просмотром и редактированием информацией, а также экспортом необходимых данных в .docx документ, откуда специалист уже сможет быстро вставлять их в основные отчеты. Программный продукт создан для автоматизации деятельности специалиста кафедры по документационному обеспечению (модуль научно-исследовательской деятельности профессорско-преподавательского состава кафедры).

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. КонсультантПлюс – надежная правовая поддержка. Справочная информация: "Размеры минимальной заработной платы в субъектах Российской Федерации" [Электронный ресурс]: РАЗМЕРЫ МИНИМАЛЬНОЙ ЗАРАБОТНОЙ ПЛАТЫ В СУБЪЕКТАХ РОССИЙСКОЙ ФЕДЕРАЦИИ (по состоянию на 10.06.2023). – Режим доступа: http://www.consultant.ru/document/cons_doc_LAW_291114/.
2. Материал из QA evolution — свободной энциклопедии [Электронный ресурс]: ЖИЗНЕННЫЙ ЦИКЛ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ. – Режим доступа: <https://qaevolution.ru/zhiznennyj-cikl-programmnogo-obespecheniya/>.
3. Материал из Википедии — свободной энциклопедии [Электронный ресурс]: IDEF3. – Режим доступа: <https://ru.wikipedia.org/wiki/IDEF3>.
4. Материал из Википедии — свободной энциклопедии [Электронный ресурс]: Автоматизированное рабочее место. – Режим доступа: https://ru.wikipedia.org/wiki/Автоматизированное_рабочее_место.
5. Материал из Википедии — свободной энциклопедии [Электронный ресурс]: Информационная система. – Режим доступа: https://ru.wikipedia.org/wiki/Информационная_система.
6. Материал из Википедии — свободной энциклопедии [Электронный ресурс]: Совокупная стоимость владения. – Режим доступа: https://ru.wikipedia.org/wiki/Совокупная_стоимость_владения.
7. Материал из информационного ресурса Microsoft [Электронный ресурс]: Системные требования для установки Windows 10. – <https://www.microsoft.com/ru-ru/windows/windows-10-specifications#primaryR2>.
8. Материал из информационного ресурса RuCore.NET [Электронный ресурс]: PythonSQLite – <https://rucore.net/p/archives/9870>.

9. Материал из информационного ресурса it-black.ru [Электронный ресурс]: IDEPyCharm. – Режим доступа: <https://it-black.ru/ide-pycharm/>.
10. Словари и энциклопедии на Академике [Электронный ресурс]: Оценка затрат на разработку программного обеспечения. – Режим доступа: <https://dic.academic.ru/dic.nsf/ruwiki/1324196>.
11. Студопедия.Нет — Информационный студенческий ресурс [Электронный ресурс]: Обоснование выбора языка программирования и IDE. – Режим доступа: https://studopedia.net/4_34511_obosnovanie-vibora-yazika-programmirovaniya-i-IDE.html.
12. Хакасский технический институт – филиал СФУ [Электронный ресурс]: Информация о ХТИ – филиале СФУ. – Режим доступа: <http://khti.sfu-kras.ru/institute/>.
13. Электронный фонд правовой и нормативно-технической документации [Электронный ресурс]: Соглашение о минимальной заработной плате в Республики Хакасия от 5 марта 2019 года, г. Абакан. – Режим доступа: <http://docs.cntd.ru/document/553222314>.

Выпускная квалификационная работа выполнена мной самостоятельно.
Использованные в работе материалы и концепции из опубликованной научной литературы и других источников имеют ссылки на них.

Отпечатано в одном экземпляре.

Библиография 13 наименований.

Один экземпляр сдан на кафедру.

«_____» _____ 2023 г.

_____ Гусев Роман Георгиевич
подпись

Федеральное государственное автономное
образовательное учреждение
высшего образования
«СИБИРСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

Хакасский технический институт – филиал ФГАОУ ВО
«Сибирский федеральный университет»

Кафедра прикладной информатики, естественно-научных и гуманитарных
дисциплин

УТВЕРЖДАЮ
Заведующий кафедрой
О. В. Папина
подпись
« 19 » июня 2023 г.

БАКАЛАВРСКАЯ РАБОТА

09.03.03 – Прикладная информатика

Автоматизированное рабочее место специалиста по документационному
обеспечению кафедры ХТИ – филиала СФУ

Руководитель М. А. Бурева 19.06.23 доцент, канд. физ-мат. наук М. А. Бурева
подпись, дата

Выпускник Р. Г. Гусев 19.06.23 Р. Г. Гусев
подпись, дата

Консультанты
по разделам:

Экономический М. А. Бурева 19.06.23 М. А. Бурева
подпись, дата

Нормоконтролер А. Н. Кадычегова 19.06.23 А. Н. Кадычегова
подпись, дата

Абакан 2023