

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ ФЕДЕРАЦИИ
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
«БАШКИРСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»

БИРСКИЙ ФИЛИАЛ БАШГУ
ФАКУЛЬТЕТ ФИЗИКИ И МАТЕМАТИКИ
КАФЕДРА ИНФОРМАТИКИ И ЭКОНОМИКИ

ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА
ПО ПРОГРАММЕ БАКАЛАВРИАТА

ЕГОРОВА АДЕЛИНА АЛЕКСАНДРОВНА

РАЗРАБОТКА АВТОМАТИЗИРОВАННОЙ СИСТЕМЫ УЧЕТА НАУЧНЫХ
ТРУДОВ ПРЕПОДАВАТЕЛЕЙ ОБРАЗОВАТЕЛЬНОЙ ОРГАНИЗАЦИИ

Выполнила:
Студентка 4 курса очной формы обучения
Направление подготовки
09.03.03 Прикладная информатика
Направленность (профиль)
Прикладная информатика в
информационной сфере

Руководитель
к.х.н., доцент

С.В Пихтовников

БИРСК – 2022

ОГЛАВЛЕНИЕ

ВВЕДЕНИЕ.....	3
ГЛАВА 1. ИССЛЕДОВАНИЕ ПРЕДМЕТНОЙ ОБЛАСТИ.....	5
1.1. Краткая характеристика организации.....	5
1.2. Структурно-функциональная диаграмма организации (финансовой структуры) «как есть» и ее описание	5
1.3. Обзор программных средств и технологий	7
1.4. Анализ затрат, выгод и рисков.....	8
Выводы по 1 главе.....	11
ГЛАВА 2. ПРОЕКТИРОВАНИЕ АВТОМАТИЗИРОВАННОЙ ИНФОРМАЦИОННОЙ СИСТЕМЫ.....	12
2.1. Разработка структурно-функциональной диаграммы по типу «Как должно быть».....	12
2.2. Проектирование диаграммы вариантов использования.....	14
2.3. Формулирование технического задания	16
2.4. Проектирование базы данных.....	19
2.5. UML диаграммы	21
2.6. Проектирование интерфейса.....	23
Выводы по 2 главе	26
ГЛАВА 3. РЕАЛИЗАЦИЯ КОМПОНЕНТОВ АИС.....	27
3.1. Реализация интерфейса.....	27
3.2. Реализация БД.....	33
3.3. Руководство пользователя.....	35
Выводы по 3 главе	43
ЗАКЛЮЧЕНИЕ	44
СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ И ЛИТЕРАТУРЫ.....	45

ВВЕДЕНИЕ

Информатизация и компьютеризация является одной из актуальных стремлений развития современной науки в последние десятилетия. Новые технологии могут оказать большую поддержку в таких областях, как планирование и анализ результатов научной деятельности, составление отчетов, обработка различной рутинной информации и т.п. В наше современное время учет научной деятельности до сих пор выполняется вручную. Западный опыт показывает нам, что переход от бумажного к электронному документообороту неминуем и является фактором, положительно сказывающимся на экономике любого субъекта.

Электронный документ уменьшает количество накладных расходов, снижает непродуктивные расходы, связанные, например, с архивным поиском документов, переводом документов в папку, отслеживанием документов. Возможность оперативного мониторинга работы в опубликованной документации, эффективное планирование времени сотрудников, централизованное хранение и хранение данных выводят учет научной деятельности на новый уровень, позволяя сотрудникам быстро получать информацию по интересующей их теме.

В данной выпускной квалификационной работе будет представлено веб-приложение автоматизированной информационной системы учета научных трудов преподавателей в образовательной организации. Проведена автоматизация работы научного сотрудника в образовательной организации.

Объектом исследования являются информационные процессы, связанные с систематизацией, анализом и статистическим учетом научных данных.

Предметом исследования является автоматизация информационных процессов, связанных с систематизацией, анализом и статистическим учетом научных данных.

Цель: разработка автоматизированной системы учёта научных трудов преподавателей для сокращения времени обработки и поиска информации.

Задачи:

- Провести анализ предметной области.
- Спроектировать компоненты автоматизированной информационной системы.
- Программно реализовать компоненты автоматизированной информационной системы.

Практическая значимость: возможность применения данной системы для автоматизации учёта научных трудов преподавателей в образовательной организации.

ГЛАВА 1. ИССЛЕДОВАНИЕ ПРЕДМЕТНОЙ ОБЛАСТИ

1.1. Краткая характеристика организации

БФ БашГУ зарегистрирован по адресу респ. Башкортостан, г. Бирск, ул. Интернациональная, д. 10, 452450. Начальник отдела информатизации образования и технического обеспечения Каскинов Н.В. Бирский филиал БашГУ основан в 1939 г. как Бирский учительский институт на базе Бирского педагогического училища. С 1952 г. преобразовался в пединститут, в 2005 г. получил статус академии. В 2012 году приказом МОиН РФ стал филиалом Башкирского государственного университета. Основным видом деятельности организации является образовательная деятельность по образовательным программам высшего образования, среднего профессионального образования, основным общеобразовательным программам, основным программам профессионального обучения, дополнительным профессиональным программам и дополнительным общеобразовательным программам.

1.2. Структурно-функциональная диаграмма организации (финансовой структуры) «как есть» и ее описание

Нотация IDEF0 предназначена для формализации и описания бизнес – процессов. С помощью средств данной нотации была создана контекстная диаграмма (Рисунок 1.1), направленная на отражение процесса работы научного сотрудника в образовательной организации.

На представленной контекстной диаграмме можно выявить следующие типы стрелок:

1. Вход – данные о научных трудах, данных о преподавателях и запрос на учёт данных
2. Выход – документы учёта.
3. Управление – нормативные документы.
4. Механизм – научный сотрудник.

Научный сотрудник регистрирует научные труды и преподавателей в картотеку, далее распределяет все карточки в алфавитном порядке, введёт учёты и создаёт отчёты. Вся работы происходит в бумажном формате, без использования какого-либо оборудования.

Расписанный процесс можно оформить следующей декомпозицией контекстной диаграммы, в которой выявлены следующие функциональные блоки (Рисунок 1.2):

1. Добавление данных о преподавателях.
2. Добавление данных о научных трудах.
3. Сортировка нужных данных.
4. Составление отчетов.

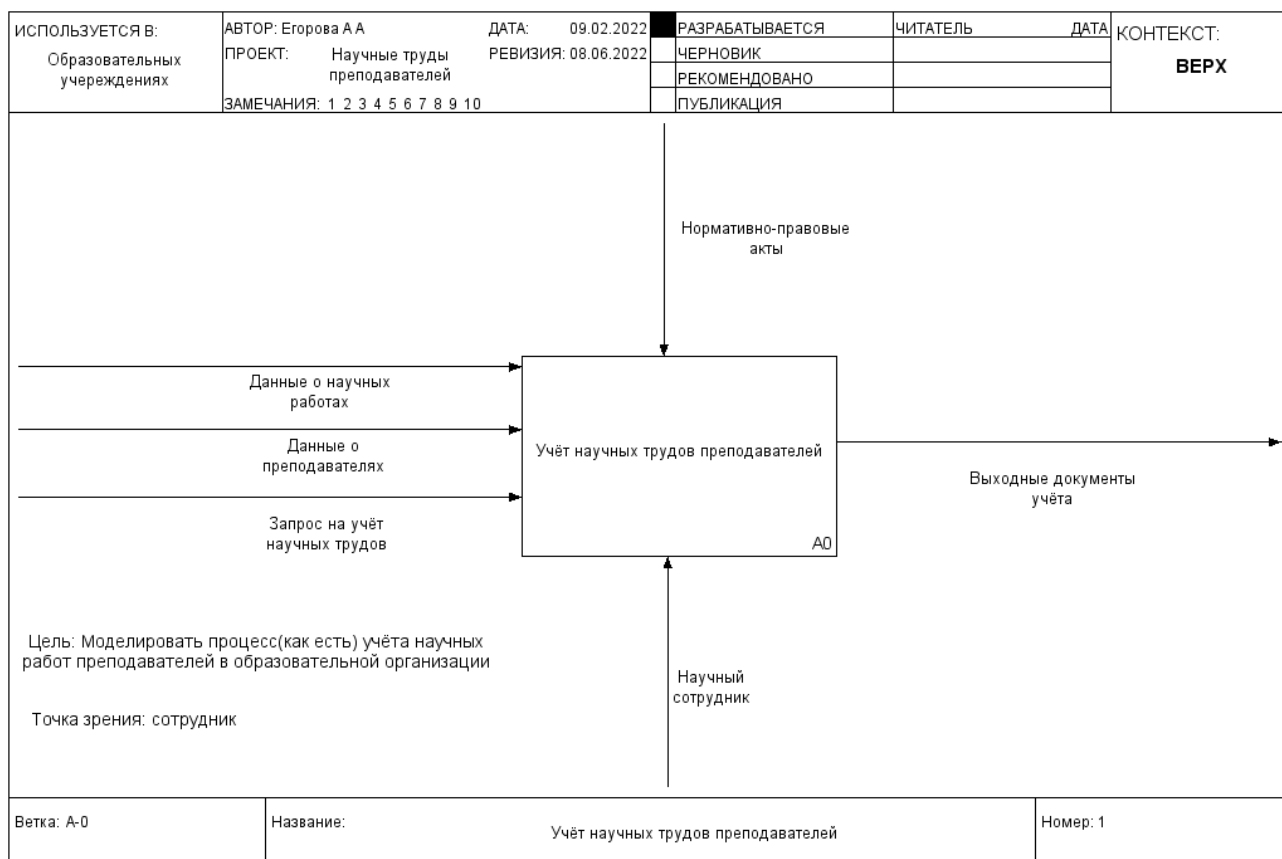


Рисунок 1.1 Контекстная диаграмма "Как есть"

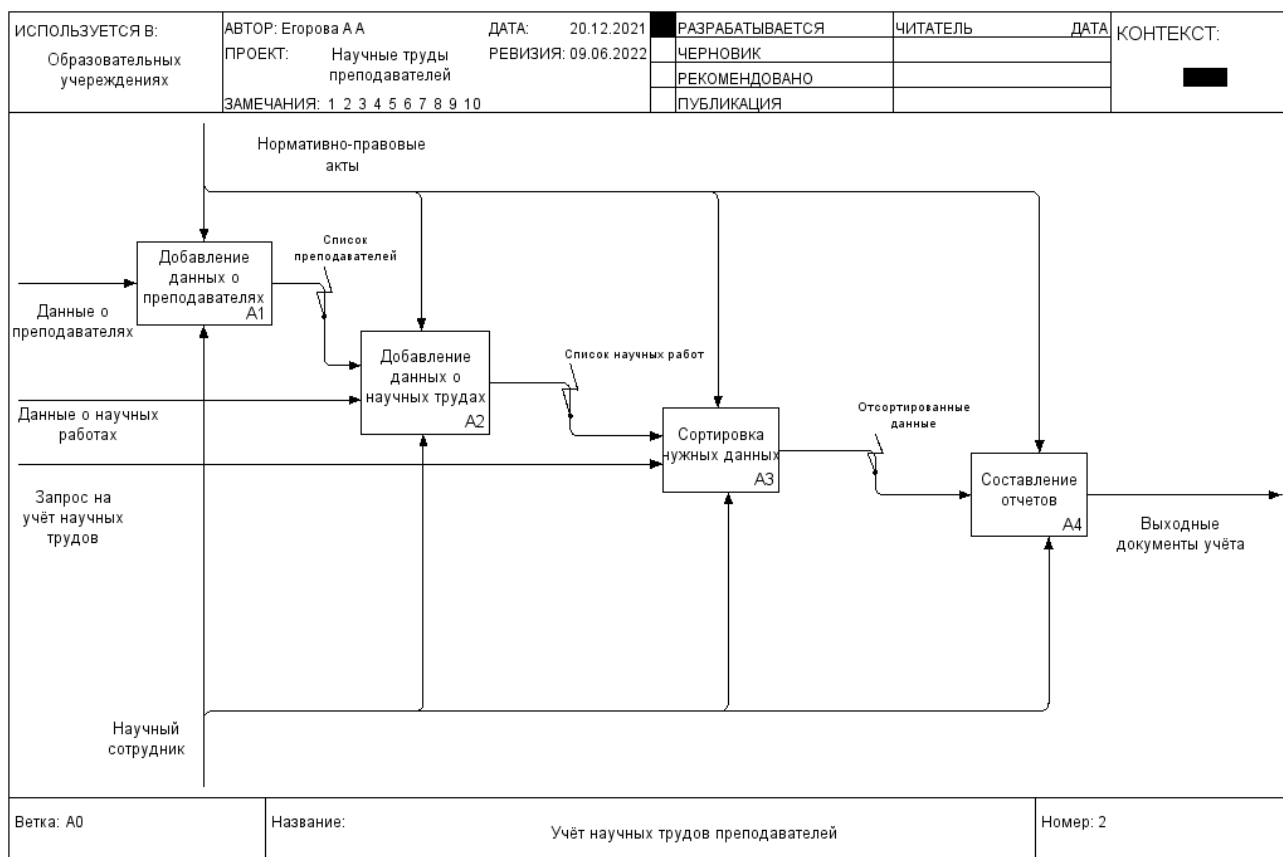


Рисунок 1.2 Результат декомпозиции контекстной диаграммы

В ходе разработки структурно-функциональной диаграммы по типу «Как есть» выявлены следующие недостатки:

1. Значительные затраты рабочего времени научного сотрудника.
2. Высокая вероятность совершения ошибки в данных.
3. Повышенная утомляемость нервной системы.

1.3. Обзор программных средств и технологий

Программное обеспечение — наряду с аппаратными средствами, важнейшая составляющая информационных технологий, включающая компьютерные программы и данные, предназначенные для решения определённого круга задач и хранящиеся на машинных носителях. Программное обеспечение представляет собой либо данные для использования в других программах, либо алгоритм, реализованный в виде последовательности инструкций для процессора. Список и описание, использованные программ при реализации АИС учёта научных трудов преподавателей, показаны в таблице 1.1.

Таблица 1.1

Прикладное программное обеспечение.

№ п/п	Программа Microsoft Office	Описание
1	2	3
1.	Microsoft Office Word	Текстовый процессор, предназначенный для создания, просмотра и редактирования текстовых документов, с локальным применением простейших форм таблично-матричных алгоритмов.
2.	Microsoft Office Excel	Программа для работы с электронными таблицами, предоставляющая возможности экономико-статистических расчетов, графические инструменты и язык макропрограммирования VBA.
3.	Visual Studio 2019	Интегрированная среда разработки ПО.

Для разработки АИС учета научных трудов преподавателей использовалась среда разработки Microsoft Visual Studio 2019.

1.4. Анализ затрат, выгод и рисков

Анализ затрат и выгод (СВА) – это процесс, используемый для измерения выгод от решения или принятия действия за вычетом затрат, связанных с этим действием. СВА обеспечивает наилучший подход к достижению выгод при сохранении экономии.

Наш проект представляет собой автоматизированную систему учета научных трудов преподавателей, созданную для обеспечения экономии времени сотрудника и сохранения целостности данных.

Трудоемкость разработки. Трудоемкость- это показатель, который складывается из общего количества времени, потраченного на каждый из этапов разработки. Разработчиком было определено следующее количество часов по каждому из этапов (Табл. 1.2).

Таблица 1.2

Трудоемкость разработки по этапам

№	Название этапа	Количество часов, ч
1.	Анализ предметной области	26
2.	Проектирование модели	98
3.	Разработка модели	80
4.	Тестирование и отладка	36
	Общая трудоемкость:	240

Из полученных данных, трудоемкость работы $T=240$ ч.

Расчет заработной платы. Заработная плата рассчитывается исходя из размера часовой заработной платы специалиста, которая в нашем случае составляет 200руб/ч., тогда:

$$\text{Заработная плата} = 200 * 240 = 48\,000 \text{ руб.}$$

На заработную плату делаются начисления в размере 30% от начисленной заработной платы.

$$\text{НЗП} = 48\,000 * 0.3 = 14\,400 \text{ руб.}$$

$$\text{Итого на разработчика: } 48\,000 \text{ руб.} + 14\,400 \text{ руб.} = 62\,400 \text{ руб.}$$

Цена оргтехники. Цена компьютера и принтера (комбоустройства) необходимого для работы научного сотрудника примерно 50 000 руб.

Расчет затрат на электроэнергию. Потребление электричества компьютером составляет примерно 220 Ватт в час. Если компьютер работает 8 часов, то расход 1,76 кВт*ч за 8 часов.

Пусть при 40-часовой неделе за год выходит 1 979 часов. Тогда за год расход на электроэнергию на компьютер менеджера с округлением 435,38 кВт*ч в год.

Цена электроэнергии 3,5 руб. за 1кВт*ч.

Итого 1523 руб. за год.

Хостинг. Для размещения проекта АИС в сети интернет необходимо приобретения хостинга. При не нагруженном проекте вполне подойдет виртуальный хостинг. Примерно 235 руб в месяц.

Покупка имени домена 200 руб.

Расчет итоговых затрат. В таблице 1.3 приводятся все затраты на разработку и эксплуатацию АИС «Учёт научных работ преподавателей в образовательной организации» за 1 год работы.

Расчеты примерные и не учитывают многих нюансов, вроде амортизационных отчислений, ремонт и обслуживание компьютера и др..

Таблица 1.3

Затраты на разработку и эксплуатацию АИС «Учёт научных работ преподавателей в образовательной организации» за 1 год

Наименование затрат	Стоимость, руб.
Заработная плата программиста	62 400
Оргтехника	50 000
Электроснабжение	1 523
Домен	200

Хостинг	2 820
Общие затраты:	116 943

В результате расчетов затраты на создание составили 116 943 рубля, что ниже, чем затраты на покупку некоторого лицензионного профессионального программного обеспечения. Тем не менее, стоимость разработки может показаться высокой для некоторых организаций.

Половину стоимости составляет компьютерная техника. Но она как правило уже есть в организациях, даже если они не используют ее для автоматизации бизнес-процессов.

Риски. Внедрение АИС несет в себе некоторые риски. В частности риски потери данных при сбое в программном обеспечении (сбой хостинга, атака на сайт, вирус). Для минимизации данных рисков необходимо регулярное копирование данных для их резервного хранения.

Выводы по 1 главе

Таким образом, в ходе первой главы описана предметная область АИС «Учёт научных работ преподавателей в образовательной организации». Описаны бизнес-процессы с помощью диаграмм IDEF0. Выявлены недостатки ручного учёта данных научных трудов, такие как значительная затрата времени сотрудника, высокая вероятность совершения ошибок, повышенная утомляемость нервной системы. Так же приведены примерные расчеты затрат и рисков от внедрения АИС «Учёт научных работ преподавателей в образовательной организации».

Приведён обзор использованных программных средств. Для создания автоматизированной информационной системы использовалась интегрированная среда разработки Microsoft Visudl Studio 2019.

ГЛАВА 2. ПРОЕКТИРОВАНИЕ АВТОМАТИЗИРОВАННОЙ ИНФОРМАЦИОННОЙ СИСТЕМЫ

2.1. Разработка структурно-функциональной диаграммы по типу «Как должно быть»

По ходу разработки бизнес-процесса была построена структурно функциональная диаграмма «Как должно быть». Добавлен новый ресурс – автоматизированная информационная система, благодаря которой сократится процент времени на обслуживание читателей, на заполнение документов и отчетности [5,11].

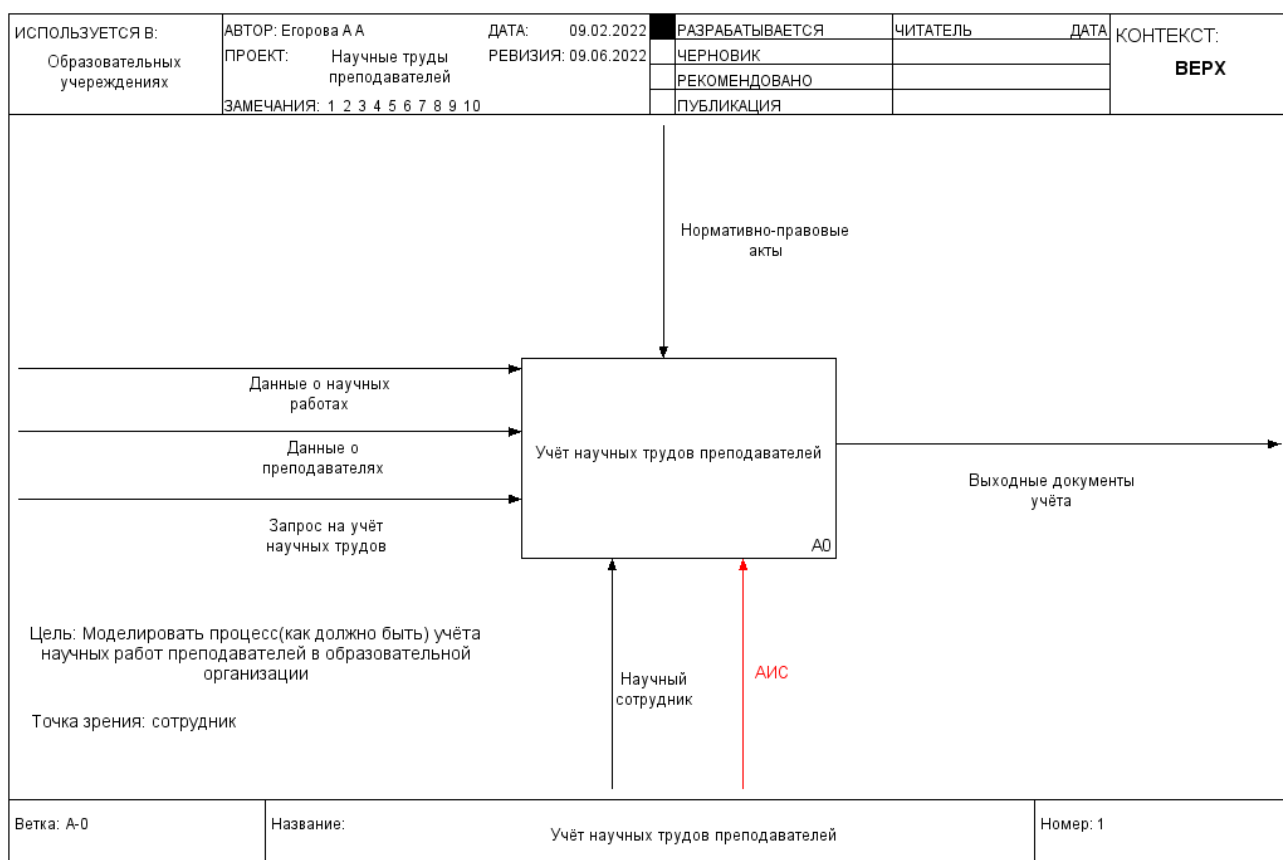


Рисунок 2.1 Контекстная диаграмма "Как должно быть"

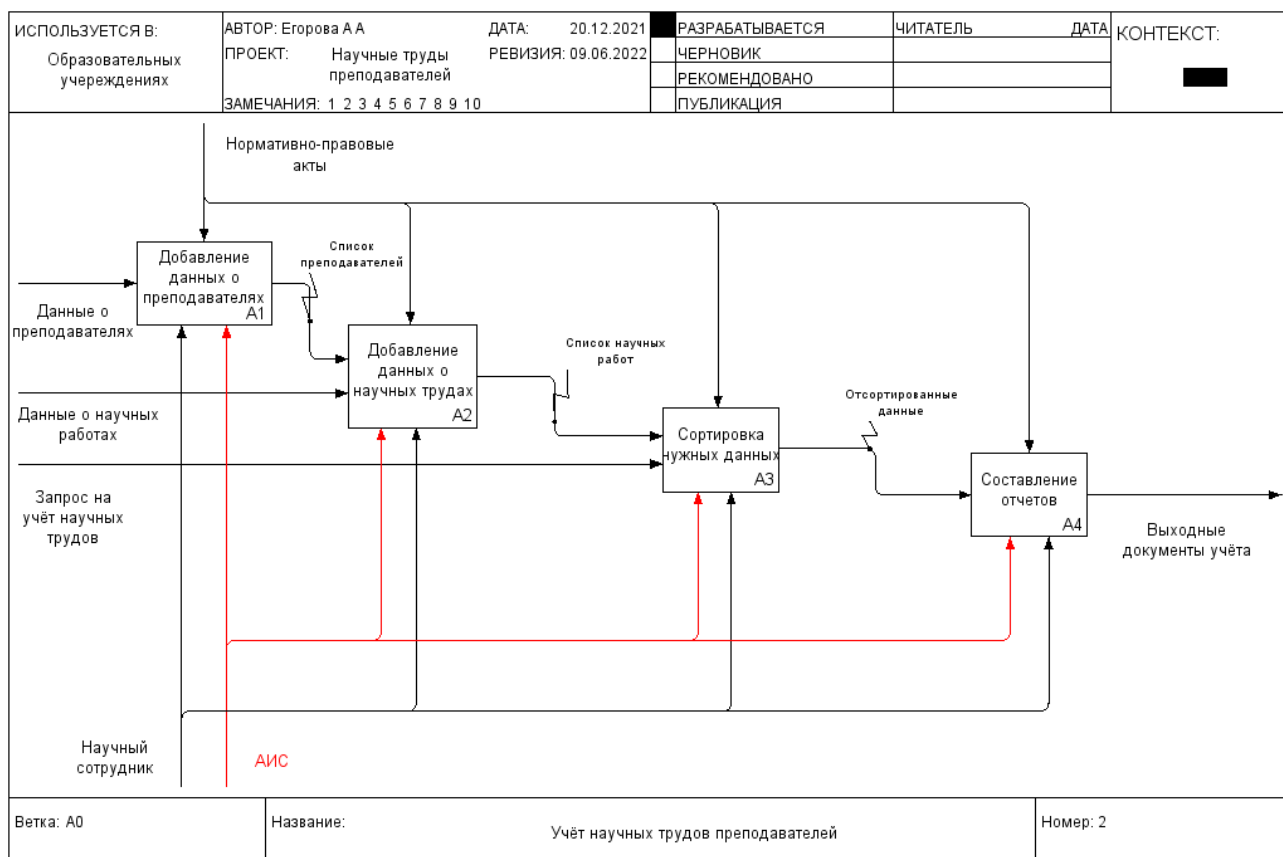


Рисунок 2.2 Декомпозиция контекстной диаграммы

Рассмотрим изменения, которые произошли в блоках модели после внедрения АИС в работу.

1. Блок «Добавление данных о преподавателях». Упрощенно добавление данных, с помощью АИС в окно добавления вводятся нужные данные, которые передаются в БД и назначается индивидуальный номер, по которому потом можно будет найти преподавателя.

2. Блок «Добавление данных о научных трудах» - добавление происходит легче из-за того, что в окне можно будет уже выбрать нужного преподавателя, данные сохраняются в БД и в будущем к ним будет проще обращаться.

3. Блок «Сортировка нужных данных» - теперь при получении запроса на отчёт, мы с помощью программы можем отсортировать данные, затратив на это минимум времени.

4. Блок «Составление отчетов» - при помощи АИС научный сотрудник теперь без труда сможет составить любой отчет в любое время рабочего дня.

2.2. Проектирование диаграммы вариантов использования

Диаграмма прецедентов или диаграмма вариантов использования — это диаграмма, на которой отражаются отношения между актерами и прецедентами системы, позволяющая описывать систему на концептуальном уровне.

Прецедент — это возможность системы, часть ее функциональности, с помощью которой актер может получить нужный ему результат. Прецедент является соответствием для отдельного сервиса системы и определяет варианты ее использования.

В диаграмме прецедентов (рисунок 2.3) представлены следующие актеры: научный сотрудник, преподаватель, администратор.

Основной функционал сайта, такие как добавление научных работ, изменение и удаление данных научных трудов, сортировка и выгрузка данных в формат Excel, доступен научному сотруднику.

Преподаватель может добавить свою научную работу, отредактировать её, а также посмотреть список имеющихся научных трудов, и если надо, скачать нужную работу.

Администратор может управлять пользователями, добавлять и удалять их.

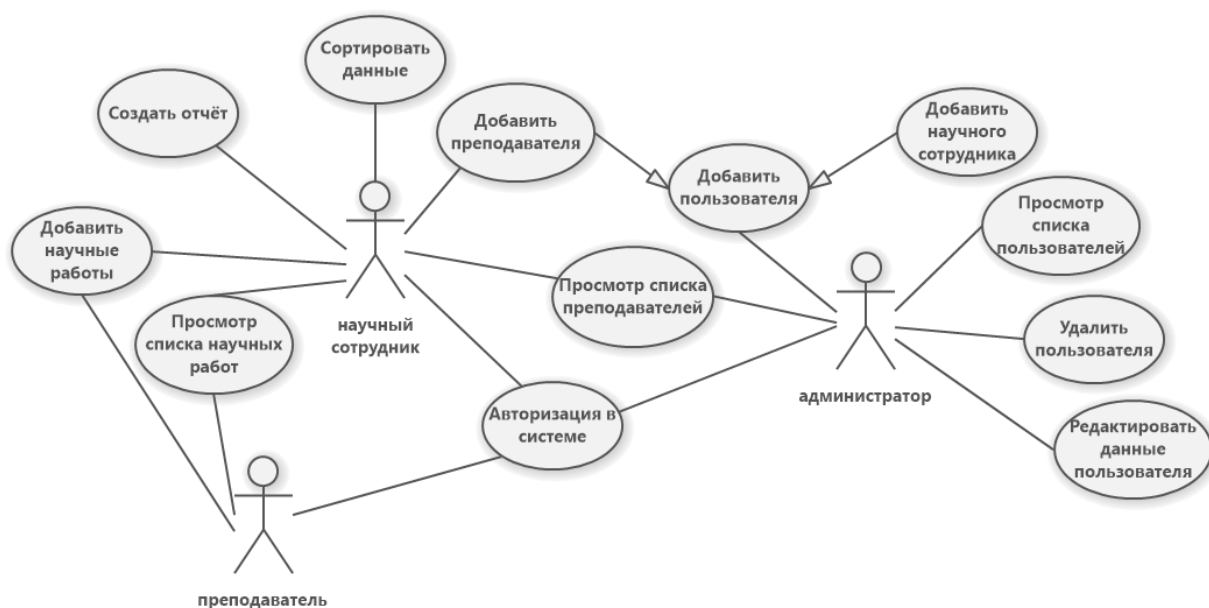


Рисунок 2.3 Диаграмм прецедентов

Описание элементов диаграммы прецедентов для администратора:

1. Администратор (актер) — главный пользователь АИС, выполняет функции по работе с пользователями.
2. Авторизация — вход в систему пользователя с определением его прав доступа.
3. Добавление пользователя — добавляет нового пользователя в систему под соответствующей ролью.
4. Удаление и редактирование пользователей — изменение данных о пользователях в БД.

Описание элементов диаграммы прецедентов для научного сотрудника:

1. Научный сотрудник (актер) — выполняет функции по работе с научными трудами.
2. Авторизация — вход в систему пользователя с определением его прав доступа.

3. Добавление преподавателя — добавляет нового преподавателя в БД, без права доступа к сайту, т.к. за добавление пользователей на сайт отвечает администратор.
4. Добавление научных трудов — добавляет новый научный труд в БД.
5. Удаление и редактирование научных трудов — изменение данных о научных трудах в БД.
6. Сортировка данных — научный сотрудник выбирает нужные данные по научным работам и сортирует их.
7. Создать отчёт — по отсортированным данным создаётся отчёт в формате Excel.

Описание элементов диаграммы прецедентов для преподавателя:

1. Преподаватель (актер) — выполняет функции по работе с научными трудами.
2. Авторизация — вход в систему пользователя с определением его прав доступа.
3. Добавление научного труда — авторизованный преподаватель может добавить работу только под своим именем.

2.3. Формулирование технического задания

Общие сведения. Полное наименование системы: автоматизированная система учета научных работ преподавателей в образовательной организации.

Краткое наименование: АИС учета научных работа преподавателей в образовательной организации.

Заказчик: БФ БашГУ, отдел информатизации

Адрес фактический: г. Бирск ул. Интернациональная, д.10

Телефон/Факс: + (34784) 4-04-99

Исполнитель: студентка 4 курса 4 группы Егорова А.А

Адрес фактический: г. Бирск ул. Интернациональная, д.122а, кв14

Телефон/Факс: +7(937)364-53-49

Работа АИС сдаётся после полного окончания работы над проектом.

Плановые сроки начала, и окончания работы по созданию системы.

Работа началась 01.02.2022 закончилась 27.05.2022.

Порядок оформления и предъявления заказчику результатов работ.

Таблица 2.1

Порядок предъявления результата работы

Наименование работ	Сроки начала и окончания работы	Отчётность
Сбор нужной информации	С 01.02.22 по 28.11.22	Информация была получена и распределена.
Создание диаграмм	С 01.03.22 по 15.03.22	Диаграммы были созданы.
Создание БД	С 15.03.22 по 30.03.22	БД «АИС учёта научных трудов преподавателей» создана
Создание web-интерфейса	С 01.04.22 по 15.05.22	Web-интерфейс на платформе ASP.NET Core с использованием шаблона MVC был создан и готов к работе с БД.
Подготовленные отчётности по проделанной работе	С 15.05.22 по 27.05.22	Отчёт готов и сдан на проверку.

Характеристика объектов автоматизации.

Таблица 2.2

Характеристики автоматизации

Структурное подразделение	Наименование процесса	Возможность автоматизации	Решение об автоматизации в ходе проекта
Отдел анализа	Анализ и учёт научных трудов преподавателей	Возможна	Будет автоматизирована

Требования к системе. В АИС предполагается выделить следующие функциональные подсистемы:

1. Подсистема сбора, обработки и загрузки данных.
2. Подсистема хранения данных.
3. Подсистема формирования и визуализации отчетности.

В основном режиме функционирования АИС должна обеспечивать:

1. Работу пользователей в режиме- 24 часа в день, 7 дней в неделю.
2. Выполнение своих функций.

Для всех технических компонентов необходимо обеспечить регулярный и постоянный контроль состояния и техническое обслуживание.

Состав и содержание работ по созданию системы. Разработка АИС осуществляется в соответствии с ГОСТ 19 и 34 серий и включает следующие стадии (этапы):

1. формирование требований к АИС.
2. разработка концепции АИС.
3. разработка технического задания.
4. эскизное проектирование.
5. техническое проектирование.
6. разработка рабочей документации.

7. ввод в действие АИС.

8. сопровождение АИС.

Требование к документированию.

Таблица 2.3

Требования к документированию

Этап	Документ
Проектирование. Разработка эскизного проекта. Разработка технического проекта.	<ul style="list-style-type: none">• Ведомость эскизного проекта• Пояснительная записка к эскизному проекту• Схема функциональной структуры
Разработка рабочей документации. Адаптация программ	<ul style="list-style-type: none">• Технологическая инструкция• Инструкция по формированию и ведению базы данных (набора данных)• Описание программ• Текст программы
Ввод в действие	<ul style="list-style-type: none">• Акт внедрения

2.4. Проектирование базы данных

Для создания информационной системы учёта научных трудов преподавателей была выбрана MS SQL Server Management Studio.

Приложение MS SQL Server Management Studio – это высокопроизводительная и мощная система управления реляционной базой данных. MS SQL Server Management Studio обеспечивает доступ ко всем типам данных и позволяет одновременно использовать несколько таблиц БД.

Несмотря на то, что MS SQL Server Management Studio является сложной системой, её с легкостью могут использовать непрофессиональные пользователи [31-33].

При разработке модели предметной области, было принято решение использовать диаграмму «сущность-связь» (ER-диаграммы) в нотации Питера Чена (рисунок 2.4). На представленной диаграмме показаны все сущности, их атрибуты и связи между ними.

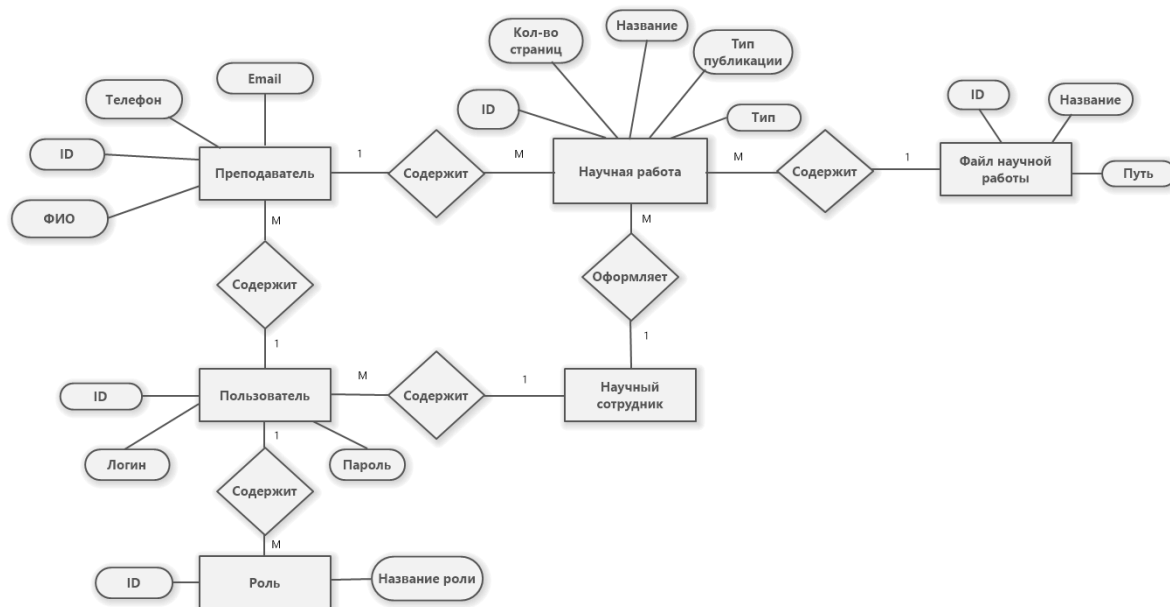


Рисунок 2.4 ER-модель в нотации Питера Чена

Физическая модель базы данных это модель данных, описанная с помощью средств конкретной СУБД. Физическая модель базы данных содержит все детали, необходимые конкретной СУБД для создания базы: наименования таблиц и столбцов, типы данных, определения первичных и внешних ключей и т.п. Даталогическая модель данных фактически является готовым заданием на создание базы данных, имея которое можно реализовать базу данных в выбранной СУБД [3,4,6,10,25].

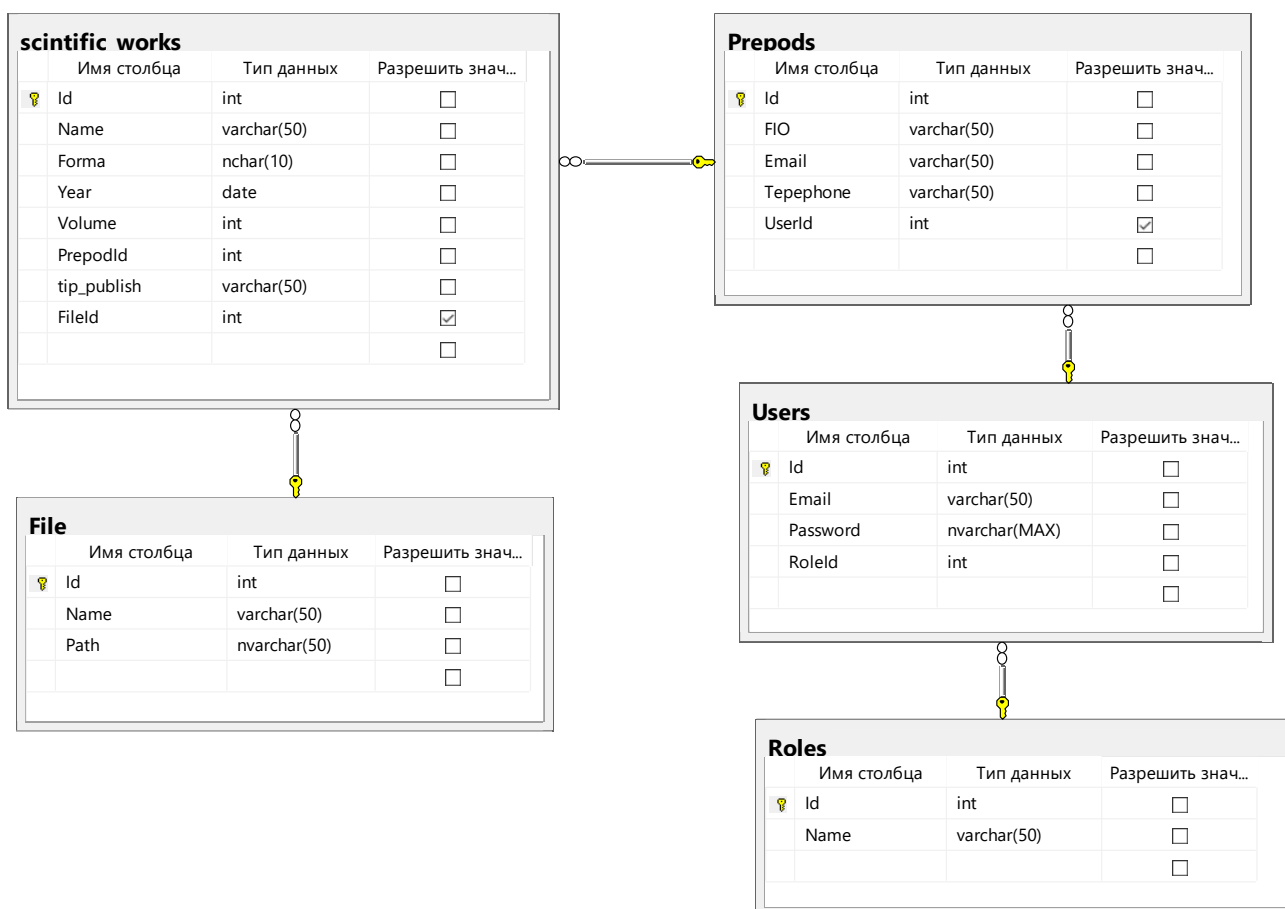


Рисунок 2.5 Даталогическая модель БД

2.5. UML диаграммы

UML (Unified Modeling Language) – это унифицированный язык, предназначенный для объектного моделирования, содержит набор инструментальных средств [7,11].

UML диаграммы:

1. Диаграмма прецедентов (Use case diagram).
2. Диаграмма классов (Class diagram).
3. Диаграмма последовательностей (Sequence diagram).
4. Диаграмма деятельности (Activity diagram).
5. Диаграмма кооперации (Collaboration diagram).
6. Диаграмма компонентов (Component diagram).
7. Диаграмма развертывания (Deployment diagram).

8. и др.

Каждая диаграмма детализирует различные представления о модели системы. UML диаграмма предназначена для визуализации, проектирования, а также документирования программного обеспечения.

Диаграмма деятельности ориентирована на разработку модели конкретного бизнес-процесса, показывает, как поток управления переходит от одной деятельности к другой.

У диаграммы деятельности может быть только одно начальное состояние, которое обозначается черным маленьким кружком, зато конечных состояний может быть множество, обозначается большим черным кружком внутри круга.

Диаграмма деятельности состоит из:

1. состояние деятельности (activity, process).
2. состояние действия (action state).
3. начальное и конечное состояние.
4. переход (transitions).
5. Ветвления.
6. разделения и слияния.
7. дорожки.

На рисунке 2.6 показана диаграмма деятельности для научного сотрудника. В начале состояния он авторизуется в системе. Если логин и пароль верный, то приступает к выбору действий, иначе снова вводит пароль и если не верный, то выходит. На диаграмме представлены 4 действия, которые может выполнить научный сотрудник. При выборе «Добавить научную работу» происходит добавление работы, в которой вводится название работы, автор работы, количество страниц и т.д. При выборе «Изменить выбранную научную работу» и «Удалить выбранную научную работу» научный сотрудник или редактирует, или удаляет работу, при это данные в БД изменяются. При «Просмотр списка научных работ» выводится список научных работ, которые можно отсортировать, и в дальнейшем экспортировать в виде отчёта в формате Excel.

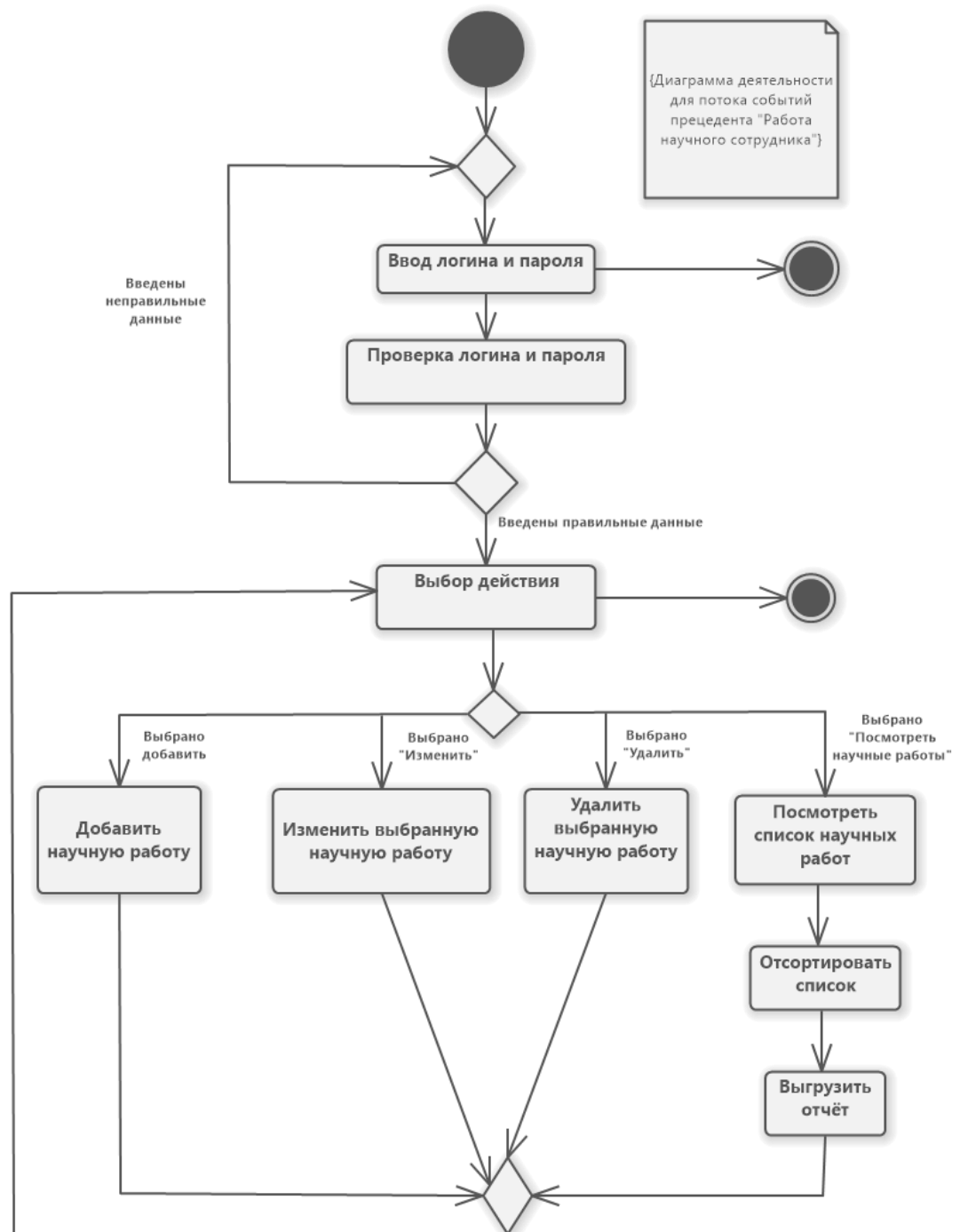


Рисунок 2.6 Диаграмма деятельности

2.6. Проектирование интерфейса

Веб-интерфейс — это совокупность средств, с помощью которых пользователи могут взаимодействовать с функционалом веб-сайта.

Распространение веб-интерфейса обусловлено с увеличением популярности сети интернета и использование повсеместно веб-браузеров [13,37,38].

Основными методами создания веб-интерфейса является использование языка разметки HTML с языком описания внешнего вида веб-страницы CSS.

Веб-интерфейс АИС учёта научных трудов преподавателей должен содержать следующие страницы:

Для всех:

- Страница «Авторизация». Появляется форма ввода логина и пароля.

В систему могут зайти только те пользователи, которым администратор назначит логины и пароли

Для научного сотрудника:

- Страница «Главная». Выводится основной функционал для научного сотрудника.

- Страница «Добавление преподавателей». Через эту страницу научных сотрудник добавляет нового преподавателя в систему. Назначать его новым пользователем может только администратор.

- Страница «Добавление научных трудов». Страница предназначена для добавление научной работы.

- Страница «Редактирование». Позволяет редактировать выбранные данные.

- Страница «Список преподавателей». На странице выводится список всех преподавателей.

- Страница «Список научных трудов». На странице выводится список всех научных работ, которые можно отсортировать, а также выгрузить в виде отчета в Excel.

Для преподавателя:

- Страница «Главная». Выводится основной функционал для преподавателя.

- Страница «Добавление научного труда». Страница предназначена для добавление научной работы. Преподаватель может добавить научную работу только под своим именем.

- Страница «Редактирование». Редактирование своих научных трудов.

- Страница «Список научных трудов». Выводится весь список научных работ, в том числе работы других преподавателей. Если файл работы был загружен, то понравившуюся научную работу можно скачать.

- Для администратора:

- Страница «Добавление пользователя». Страница в котором отображены основные поля для добавления преподавателя в систему.

- Страница «Редактирование». Позволяет редактировать нужного пользователя, изменяя его логин или пароль, а также удаляя его из системы.

- Страница «Список пользователей». Выводится список всех пользователей.

Макет веб-страницы — это общая схема страницы и расположенных на ней элемент. На рисунке 2.7 изображён макет главной страницы для АИС учета научных трудов преподавателей, выполненный в Adobe Illustrator.



Рисунок 2.7 Макет сайта

Этот макет содержит следующие элементы:

- Шапка.

- Схему глобальной навигации — элементы навигации, которые содержатся на всех страницах сайта.
- Основной блок.
- Подвал.

Выводы по 2 главе

В ходе работы над данной главой построены: диаграмма деятельности, ER-диаграмма в нотации Питера Чена, диаграмма прецедентов. Разработана физическая модель базы данных. Определена и построена диаграмма IDEF0 «Как должно быть». Спроектирован веб-интерфейс для АИС.

ГЛАВА 3. РЕАЛИЗАЦИЯ КОМПОНЕНТОВ АИС

3.1. Реализация интерфейса

Веб-интерфейс реализован с помощью встроенного в платформу ASP.NET CORE паттерна MVC.

Концепция паттерна MVC предполагает разделение приложения на три компонента: model, view, controller(MVC).

- **Модель (model):** описывает используемые в приложении данные, а также логику, которая связана непосредственно с данными, например, логику валидации данных.
- **Представление (view):** отвечают за визуальную часть или пользовательский интерфейс в виде html-страниц, через который пользователь взаимодействует с приложением. Также представление может содержать логику, связанную с отображением данных. В то же время представление не должно содержать логику обработки запроса пользователя или управления данными.
- **Контроллер (controller):** представляет центральный компонент MVC, который обеспечивает связь между пользователем и приложением, представлением и хранилищем данных. Он содержит логику обработки запроса пользователя. Контроллер получает вводимые пользователем данные и обрабатывает их. И в зависимости от результатов обработки отправляет пользователю определенный вывод, например, в виде представления, наполненного данными моделей [1,2,9,15-18,39].

Весь пользовательский интерфейс состоит из представлений, написанных на HTML коде с использованием каскадных таблиц стилей CSS и фреймворка Bootstrap . Внизу представлен пример кода представления главной страницы администратора:

```
@{
    ViewData["Title"] = "Home Page";
}
<html>
<head>
    <title>Администратор</title>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1">
```

```

    <link rel="stylesheet" type="text/css" href="~/css/blockGlav.css">
</head>
<body>

    <div class="post-wrap">
        <div class="post-item">
            <div class="item-content">

                <div class="item-body">
                    <h3>Добро пожаловать!</h3>
                    <p>Вы зашли под ролью администратор.</p>
                    <br />
                    <br />
                </div>
                <div class="item-footer">

                    </div>
                </div>
            </div>
            <div class="post-item">
                <div class="item-content">

                    <div class="item-body">
                        <h3>Добавить преподавателя.</h3>
                        <p>Добавить нового преподавателя в систему.</p>
                    </div>
                    <div class="item-footer">
                        <a asp-controller="Home" asp-action="Create"
class="link"><span>Добавить</span></a>
                    </div>
                </div>
            </div>
            <div class="post-item">
                <div class="item-content">

                    <div class="item-body">
                        <h3>Добавить пользователя*.</h3>
                        <p>*Добавить пользователя преподаватель</p>
                        <br />
                    </div>
                    <div class="item-footer">
                        <a asp-controller="Home" asp-action="CreateUser"
class="link"><span>Добавить</span></a>
                    </div>
                </div>
            </div>
            <div class="post-item">
                <div class="item-content">

```

```

        <div class="item-body">
            <h3>Добавить пользователя**.</h3>
            <p>**.Добавить пользователя научный сотрудник.</p>
        </div>
        <div class="item-footer">
            <a asp-controller="Home" asp-action="CreateUserSotr"
class="link"><span>Добавить</span></a>
        </div>
    </div>
</div>

<div class="post-item">
    <div class="item-content">

        <div class="item-body">
            <h3>Список пользователей.</h3>
            <p>Посмотреть список всех пользователей.</p>
            <br />
        </div>
        <div class="item-footer">
            <a asp-controller="Home" asp-action="SpisokUser"
class="link"><span>Перейти к списку</span></a>
        </div>
    </div>
</div>

</div>
</body>
</html>

```

Для передачи данных из представления в контроллер используются Tag-хелперы. Tag-хелперы представляют из себя функционал, предназначенный для генерации HTML-страницы. В представлении Tag-хелперы выглядят как обычные html-элементы, одно при работе приложения они обрабатываются на стороне сервера и в дальнейшем преобразуются в html-элемент [37,38,35,20,28]. На рисунке 3.2 показан пример tag-хелпера. Tag-хелпер «asp-controller» назначает контроллер, который будет использоваться для создания URL-адреса, в показанном примере контроллер «Home». Атрибут «asp-action» представляет имя действия контроллера [40]. Пример использования:

```

<form asp-action="CreatePubl" asp-controller="home" method="post"
enctype="multipart/form-data">

```

В проекте папка Views является основной папкой, где хранятся все представления. Добавление нового представления происходит с добавления новой папки в Views. По стандарту название папки должно соответствовать с используемым в дальнейшем контроллером. Далее в папку добавляется Razor-представление. На рисунке 3.1 изображены все имеющиеся папки в Views.

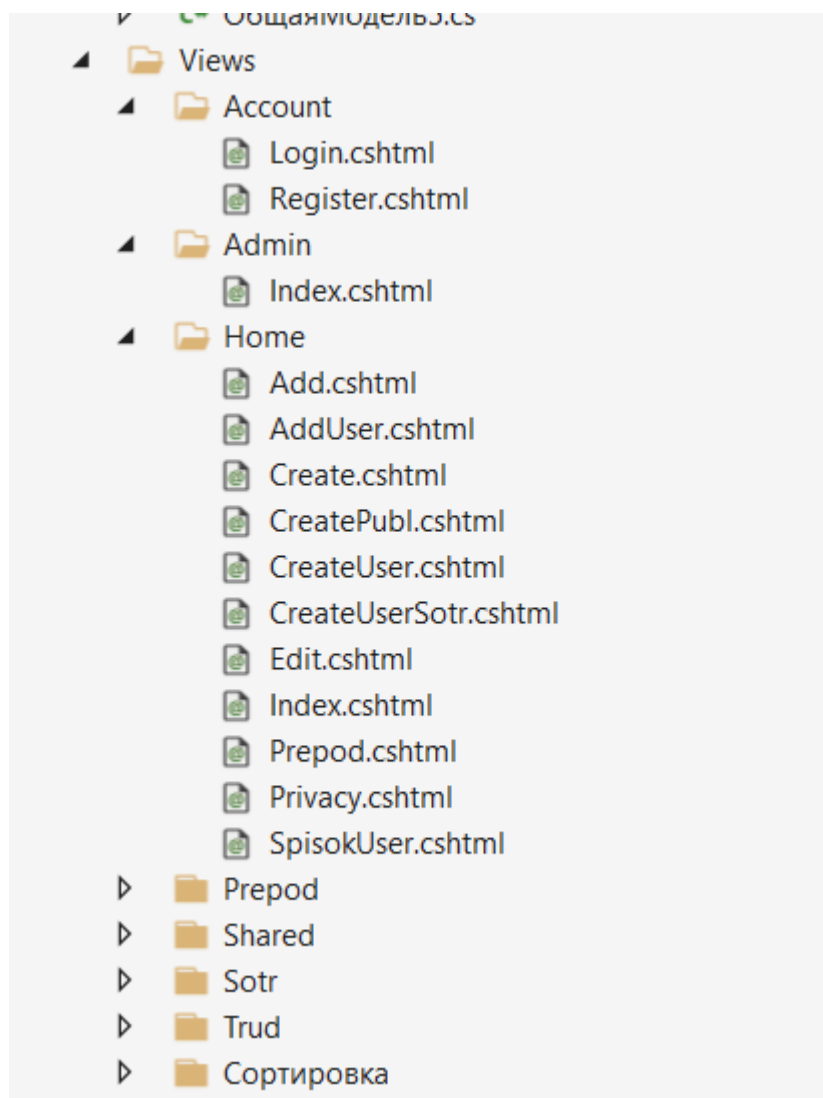


Рисунок 3.1 Список представлений

Главное оформление, панель навигации, подключение скриптов и стилей сайта находятся в файле `_Layout.cshtml`. Он состоит из двух частей шапки `<head>`, `<header>` и тела `<body>`.

В < head > файла _Layout.cshtml описано подключение всех необходимых компонентов для реализации интерфейса [19,20,22,30]. Ниже представлен фрагмент кода подключения компонентов:

```
<head>
    <meta charset="utf-8" />
    <meta name="viewport" content="width=device-width, initial-
scale=1.0" />
    <title>@ViewData["Title"] - Научный труды
преподавателей</title>
    <link rel="stylesheet"
href="~/lib/bootstrap/dist/css/bootstrap.min.css" />
    <link rel="stylesheet" href="~/css/site.css" />
</head>
```

В <header> реализована панель навигации сайта:

```
<header>
    <nav class="navbar navbar-expand-sm navbar-toggleable-sm
navbar-light bg-white border-bottom box-shadow mb-3">

        <div class="container">
            @if (User.IsInRole("Admin") == true)
            {<a class="navbar-brand" asp-area="" asp-
controller="Admin" asp-action="Index">Научные труды</a>}
            @if (User.IsInRole("Науч_сотр") == true)
            {<a class="navbar-brand" asp-area="" asp-
controller="Sotr" asp-action="Index">Научные труды</a>}
            @if (User.IsInRole("Преподаватель") == true)
            {<a class="navbar-brand" asp-area="" asp-
controller="Prepod" asp-action="Index">Научные труды</a>}
            <button class="navbar-toggler" type="button" data-
toggle="collapse" data-target=".navbar-collapse" aria-
controls="navbarSupportedContent"
                aria-expanded="false" aria-label="Toggle
navigation">

                <span class="navbar-toggler-icon"></span>
            </button>
            <div class="navbar-collapse collapse d-sm-inline-
flex justify-content-between">
                <ul class="navbar-nav flex-grow-1"
style="background-color: #E0D4FF; font-weight: 600;">
                    @*<li class="nav-item">
                        <a class="nav-link text-dark" asp-area="" asp-
controller="Home" asp-action="Index">Главная</a>
                    </li>*@

                    @if (User.IsInRole("Admin") == true)
                    {
                        <li class="nav-item">
```

```

        <a class="nav-link text-dark" asp-
area="" asp-controller="Home" asp-action="Index">Список
преподавателей</a>
    </li>
    <li class="nav-item">
        <a class="nav-link text-dark" asp-
area="" asp-controller="Home" asp-action="Create">Добавить
преподавателя</a>
    </li>
    <li class="nav-item">
        <a class="nav-link text-dark" asp-
area="" asp-controller="Home" asp-action="AddUser">Добавить
пользователя</a>
    </li>
    <li class="nav-item">
        <a class="nav-link text-dark" asp-
area="" asp-controller="Home" asp-action="SpisokUser">Список
пользователей</a>
    </li>
}

@if (User.IsInRole("Науч_сотр") == true)
{
    <li class="nav-item">
        <a class="nav-link text-dark" asp-
area="" asp-controller="Home" asp-action="Index">Список
преподавателей</a>
    </li>
    <li class="nav-item">
        <a class="nav-link text-dark" asp-
area="" asp-controller="Trud" asp-action="Index">Научные труды</a>
    </li>
    <li class="nav-item">
        <a class="nav-link text-dark" asp-
area="" asp-controller="Sotr" asp-action="AddPublish">Добавить
научн.труд</a>
    </li>
    <li class="nav-item">
        <a class="nav-link text-dark" asp-
area="" asp-controller="Сортировка" asp-action="Index">Сортировка</a>
    </li>
}
@if (User.IsInRole("Преподаватель") ==
true)
{
    <li class="nav-item">
        <a class="nav-link text-dark" asp-
area="" asp-controller="Prepod" asp-action="Index">Главная</a>
    </li>
    <li class="nav-item">

```



```

        <a class="nav-link text-dark" asp-
area="" asp-controller="Prepod" asp-action="AddPublish">Добавить
науч.труд</a>
    </li>
    <li class="nav-item">
        <a class="nav-link text-dark" asp-
area="" asp-controller="Prepod" asp-action="SpispkTrud">Мои
научн.труды</a>
    </li>
    <li class="nav-item">
        <a class="nav-link text-dark" asp-
area="" asp-controller="Trud" asp-action="Index">Список науч.труд</a>
    </li>
    }

    <li class="nav-item">
        <a class="nav-link text-dark" asp-
area="" asp-controller="Account" asp-action="Logout">Выйти</a>
    </li> </ul> </div> </div> </nav>

</header>

```

В теле <body> происходит вызов основной части страницы (Рисунок 3.6.) представляющий собой cshtml-страницу(представление) и подвала:

```

<div class="container">
    <main role="main" class="pb-3">
        @RenderBody()
    </main>
</div>

<footer class="border-top footer text-muted" style="backdrop-
filter: blur(2px);">
    <div class="container" style="color:black;">
        &copy; 2022 - Научные труды преподавателей, Егорова А.А
- <a asp-area="" asp-controller="Home" asp-action="Privacy">Privacy</a>
    </div>
</footer>
<script src="~/lib/jquery/dist/jquery.min.js"></script>
<script
src="~/lib/bootstrap/dist/js/bootstrap.bundle.min.js"></script>
<script src="~/js/site.js" asp-append-version="true"></script>
@await RenderSectionAsync("Scripts", required: false)

```

3.2. Реализация БД

Для подключения БД используется технология доступа к базам данных от Microsoft Entity Framework, которая автоматически связывает обычные классы языка C# с таблицами в базе данных. Entity Framework Core нацелена в первую очередь на работу с СУБД MS SQL Server, однако поддерживает также и ряд

других СУБД. В данном случае мы будем работать с базами данных в MS SQL Server [8,14,21,29,31,34-36].

Для взаимодействия с базой данных через EF используется контекст данных – класс, унаследованный от класса Microsoft.EntityFrameworkCore.DbContext [25,26]. В проекте был создан контекст данных WorkContext:

```
public class WorkContext : DbContext
{
    public DbSet<Prepod> Prepods { get; set; }
    public DbSet<Role> Roles { get; set; }
    public DbSet<User> Users { get; set; }
    public DbSet<FileModel> Files { get; set; }
    public DbSet<scientific_work> scientific_Works { get; set; }

    public WorkContext(DbContextOptions<WorkContext> options)
        : base(options)
    {
        Database.EnsureCreated();
    }
}
```

Чтобы подключиться к базе данных, в appsettings.json задали параметры подключения:

```
"ConnectionStrings": {
  "DefaultConnection": "Server=(localdb)\\mssqllocaldb;Database=sci_work;Trusted_Connection=True;MultipleActiveResultSets=true"
}
```

В конце изменяем в файле Startup.cs метод ConfigureServices(). В методе передаём строку подключения из конфигурации и добавляем контекст в качестве сервиса в приложение:

```
public void ConfigureServices(IServiceCollection services)
{
    string connection =
    Configuration.GetConnectionString("DefaultConnection");
    services.AddDbContext<WorkContext>(options =>
    options.UseSqlServer(connection));
    services.AddControllersWithViews();

    services.AddAuthentication(CookieAuthenticationDefaults.AuthenticationScheme)
        .AddCookie(options => //CookieAuthenticationOptions
        {
```

```
options.LoginPath = new  
Microsoft.AspNetCore.Http.PathString("/Account/Login");  
});  
}
```

3.3. Руководство пользователя

В ходе выполнения выпускной квалификационной работы разработана и реализована автоматизированная информационная система учета научных трудов преподавателей в образовательной организации. Разработан веб-интерфейс, который предоставляет пользовательский интерфейс для взаимодействия с сервисом или устройством посредством протокола HTTP и веб-браузера.

Для обеспечения надежности данного web-приложения реализовано:

- Разработан удобный интерфейс, для того чтобы даже человек, не пользующийся компьютером достаточно долго, мог быстро разобраться на сайте.
- Дизайн выполнен достаточно просто, не в ущерб навигации по сайту, что хорошо сказывается на нагрузке ресурсов компьютера и каналов связи. Данный момент позволяет увеличить скорость работы сайта.
- Сайт включает в себя ссылки, что помогает пользователю проще ориентироваться на сайте и переходить на нужный ему узел, в том числе на первоначальный.

При открытии web-приложения нас встречает окно регистрации (Рисунок 3.2). Пароль и логин для входа получают пользователи от администратора.

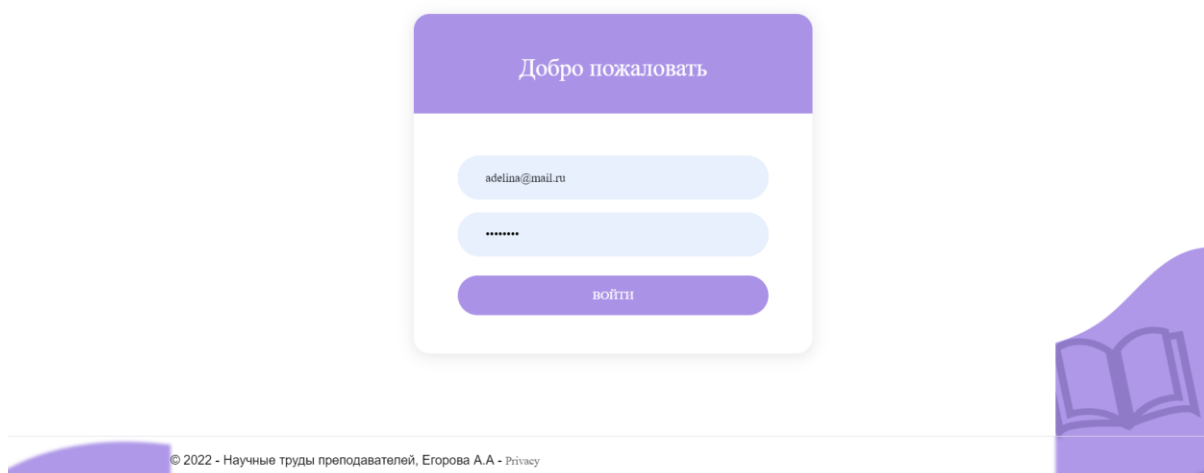


Рисунок 3.2 Окно авторизации

Вход от имени администратора. На главной странице имеются функциональные блоки администратора (Рисунок 3.3).

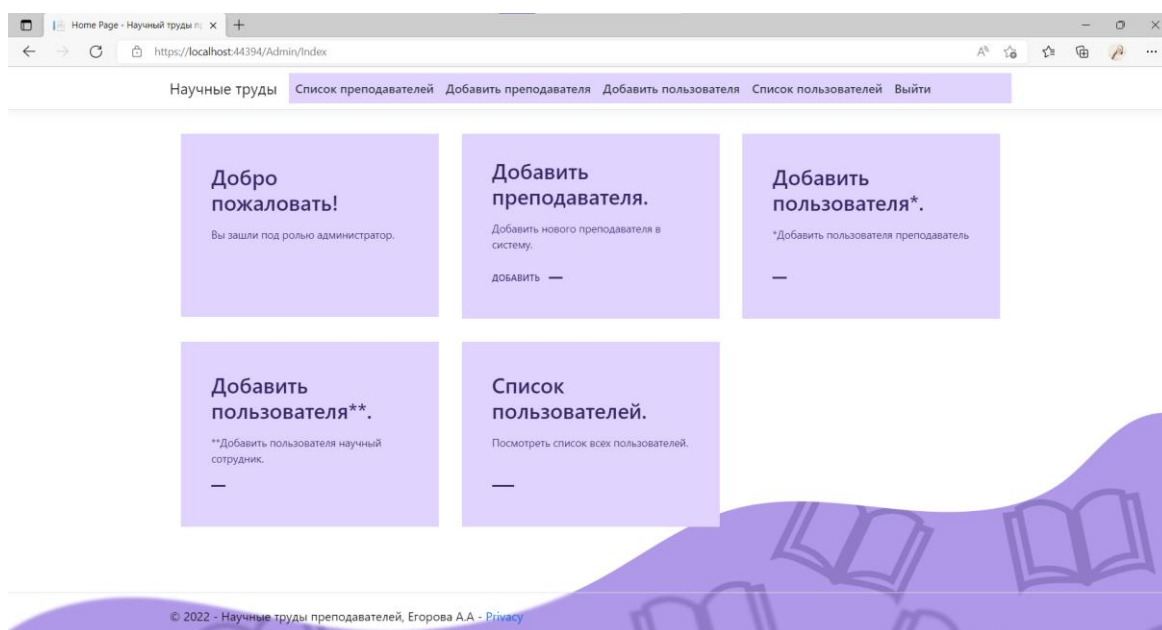


Рисунок 3.3 Главная страница администратора

Администратор может добавлять преподавателей (Рисунок 3.4), добавлять пользователя в зависимости от роли (Рисунок 3.5), просматривать список пользователей (Рисунок 3.6), список преподавателей, а так же удалять их.

Научные труды

Список преподавателей

Добавить преподавателя

Добавить пользователя

Список пользователей

Выйти

ФИО

email

телефон

+7() - -

Сохранить

Назад

© 2022 - Научные труды преподавателей, Егорова А.А - Privacy

Рисунок 3.4 Страница добавление преподавателя

Научные труды

Список преподавателей

Добавить преподавателя

Добавить пользователя

Список пользователей

Выйти

email

пароль

Роль

Научный сотрудник

Сохранить

Назад

© 2022 - Научные труды преподавателей, Егорова А.А - Privacy

Рисунок 3.5 Страница добавление пользователя

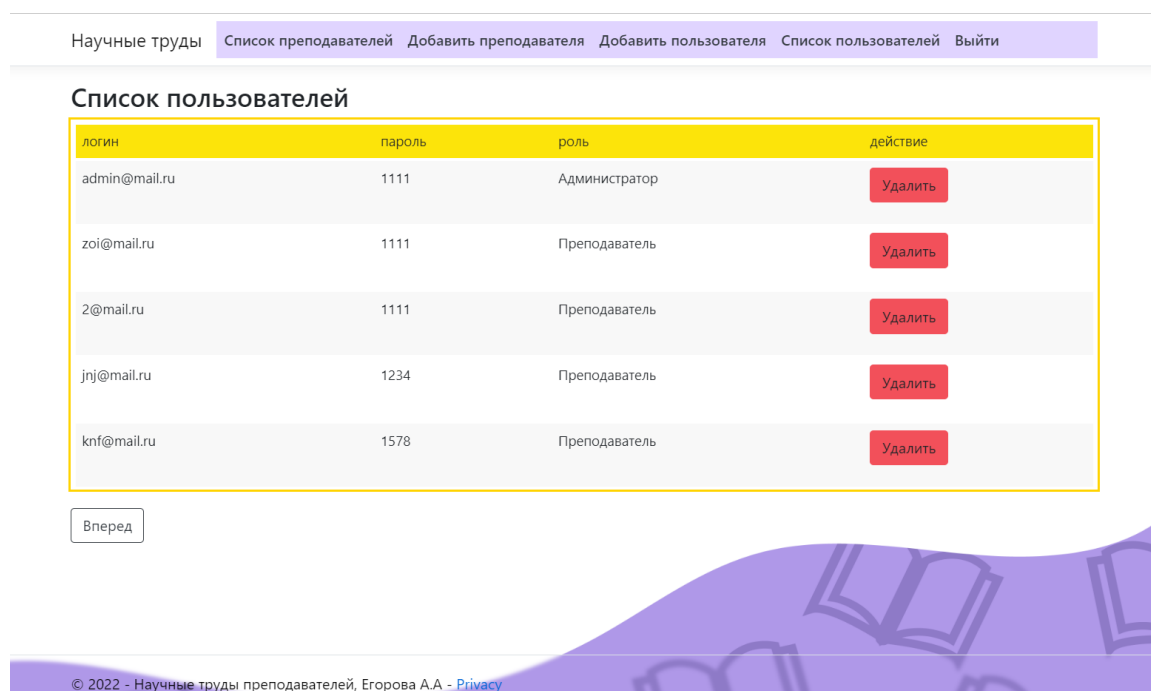


Рисунок 3.6 Список пользователей

Вход от имени научного сотрудника. На главной странице имеются функциональные блоки научного сотрудника (Рисунок 3.7).

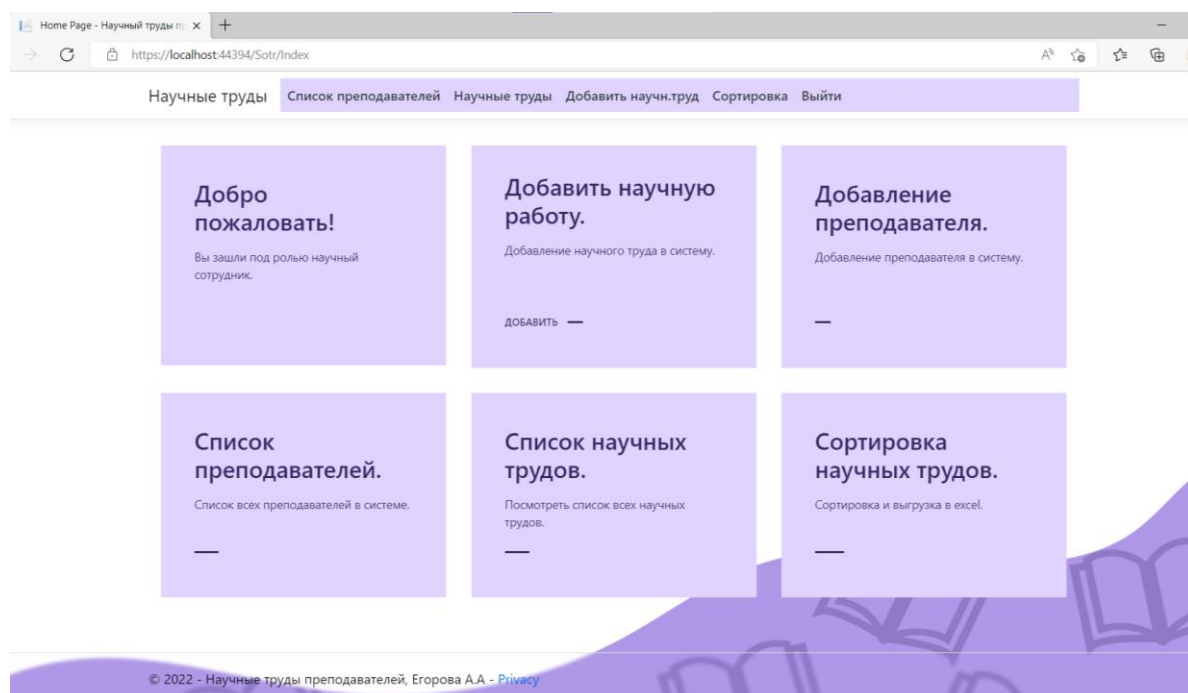


Рисунок 3.7 Главная страница научного сотрудника

Научный сотрудник может просматривать список преподавателей (Рисунок 3.8), добавлять научную работу (Рисунок 3.9), просматривать список научных работ (Рисунок 3.10), сортировать научные труды (Рисунок 3.11).

Научные труды	Список преподавателей	Научные труды	Добавить научн.труд	Сортировка	Выйти
---------------	-----------------------	---------------	---------------------	------------	-------

Список преподавателей				
ФИО	email	телефон	редактировать	удалить
Каллимуллина Ольга Айдаровна	olha@mail.ru	+7(937)744-35-86	Редактировать	Удалить
Егорова Наталья Владимировна	2@mail.ru	+7(937)839-65-85	Редактировать	Удалить
Смирнова Ольга Владимировна	cat@mail.ru	+7(937)594-65-31	Редактировать	Удалить
Камалова Диана Ильшатовна	di@mail.ru	+7(937)594-65-53	Редактировать	Удалить
Авахдеева Лена Ринатовна	lena@mail.ru	+7(937)594-65-21	Редактировать	Удалить
Галеев Олег Андреевич	oleg@gmail.ru	+7(937)594-65-34	Редактировать	Удалить
Басырова Алеся Даниловна	olec@mail.ru	+7(937)594-65-63	Редактировать	Удалить
Никифоров Пётр Олегович	petr@mail.ru	+7(937)594-27-90	Редактировать	Удалить

Рисунок 3.8 Страница списка преподавателей

На странице добавления научных трудов вводится название научной работы, выбирается тип публикации через выпадающий список, далее также добавляется фамилия имя отчество преподавателя, форма публикации, год и количество страниц. Самым последним по желанию добавляется файл научной работы.

Научные труды	Список преподавателей	Научные труды	Добавить научн.труд	Сортировка	Выйти
---------------	-----------------------	---------------	---------------------	------------	-------

Название научной работы

РИНЦ

Каллимуллина Ольга Айдаровна

Форма

Год

Кол-во страниц

Выбор файла

Не выбран ни один файл

Сохранить

Рисунок 3.9 Добавление научной работы

Список научных трудов

Название	Формат	Год	Объем	Тип публикации	Автор	Скачать		
Наука о людях и компьютерах	Научно-технический отчет	2022	190	РИНЦ	Егорова Наталья Владимировна	Скачать файл	Редактировать	Удалить
Как компьютеры влияют на мозг человека?	Тезисы	2022	169	Scopus	Авхадеева Лена Ринатовна	Скачать файл	Редактировать	Удалить
Синергетические принципы построения автопилотов	Статья	2008	180	BAK	Егорова Наталья Владимировна	Скачать файл	Редактировать	Удалить
Частотный квадруплексер на подвешенной подложке	Монография	2009	2009	BAK	Смирнова Ольга Владимировна	Скачать файл	Редактировать	Удалить
Синергетический синтез антихаотических законов управления системой Ресслера	Доклад	2004	48	Scopus	Камалова Диана Ильшатовна	Скачать файл	Редактировать	Удалить
Использование информационно – коммуникационных технологий при формировании вычислительных навыков	Монография	2021	150	Web_Of_Scinces	Галеев Олег Андреевич	Скачать файл	Редактировать	Удалить
Трассировка в канале методом муравьиной колонии	Статья	2008	190	BAK	Каллимуллина Ольга Айдаровна	Скачать файл	Редактировать	Удалить

Рисунок 3.10 Список научных трудов

На странице сортировки можно сортировать данные по названию, году публикации, преподавателю. Далее полученные данные можно экспортировать в Excel.

Название: Год: Преподаватель: [Фильтр](#) [Экспортировать в Excel](#)

Список научных трудов

Название	Формат	Год	Объем	Автор
Наука о людях и компьютерах	Научно-технический отчет	2022	190	Егорова Наталья Владимировна
Как компьютеры влияют на мозг человека?	Тезисы	2022	169	Авхадеева Лена Ринатовна
Синергетические принципы построения автопилотов	Статья	2008	180	Егорова Наталья Владимировна
Частотный квадруплексер на подвешенной подложке	Монография	2009	2009	Смирнова Ольга Владимировна
Синергетический синтез антихаотических законов управления системой Ресслера	Доклад	2004	48	Камалова Диана Ильшатовна
Использование информационно – коммуникационных технологий при формировании вычислительных навыков	Монография	2021	150	Галеев Олег Андреевич
Трассировка в канале методом муравьиной колонии	Статья	2008	190	Каллимуллина Ольга Айдаровна
АИС учета научных трудов преподавателей	Доклад	2022	45	Егорова Наталья Владимировна

Кол-во записей: 8

Рисунок 3.11 Сортировка списка научных работ

Вход от имени преподавателя. На главной странице располагаются функциональные блоки преподавателя (Рисунок 3.12).

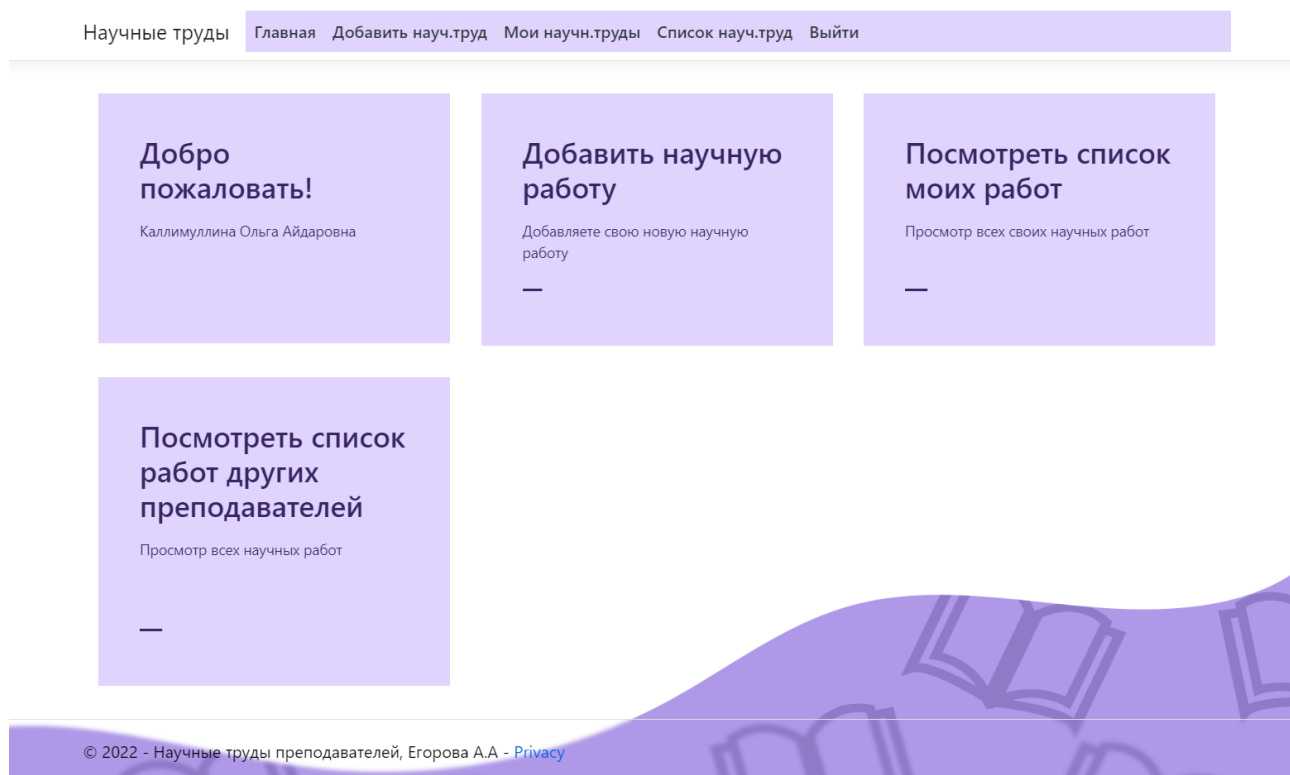


Рисунок 3.12 Главная страница преподавателя

Преподаватель может просматривать список научных трудов (Рисунок 3.13), просматривать свои научные труды (Рисунок 3.14), добавлять свои научные работы (Рисунок 3.15).

Список научных трудов

Название	Формат	Год	Объем	Тип публикации	Автор	Скачать
Наука о людях и компьютерах	Научно-технический отчет	2022	190	РИНЦ	Егорова Наталья Владимировна	Скачать файл
Как компьютеры влияют на мозг человека?	Тезисы	2022	169	Scopus	Авхадеева Лена Ринатовна	Скачать файл
Синергетические принципы построения автопилотов	Статья	2008	180	BAK	Егорова Наталья Владимировна	Скачать файл
Частотный квадруплексер на подвешенной подложке	Монография	2009	2009	BAK	Смирнова Ольга Владимировна	Скачать файл
Синергетический синтез антихаотических законов управления системой Ресслера	Доклад	2004	48	Scopus	Камалова Диана Ильшатовна	Скачать файл
Использование информационно – коммуникационных технологий при формировании вычислительных навыков	Монография	2021	150	Web_Of_Scinces	Галеев Олег Андреевич	Скачать файл
Трассировка в канале методом муравьиной колонии	Статья	2008	190	BAK	Каллимуллина Ольга Айдаровна	Скачать файл
АИС учета научных трудов преподавателей	Доклад	2022	45	BAK	Егорова Наталья Владимировна	Скачать файл

Рисунок 3.13 Список научных работ

Список научных трудов

Название	Формат	Год	Объем	Тип публикации	Автор	Скачать		
Трассировка в канале методом муравьиной колонии	Статья	2008	190	BAK	Каллимуллина Ольга Айдаровна	Скачать файл	Редактировать	Удалить

[Назад](#)

Рисунок 3.14 Список своих научных работ

Название научной работы	
РИНЦ	Каллимуллина Ольга Айдаровна
Форма	Год
Кол-во страниц	Выбор файла Не выбран ни один файл
<button>Сохранить</button>	

Рисунок 3.15 Добавление своей научной работы

Выводы по 3 главе

В третьей главе выпускной квалификационной работы подробно описана реализация интерфейса веб-приложения учета научных трудов, а также реализация подключения интерфейса к базе данных.

Написано подробное руководство пользователя, предназначенное для упрощения освоения веб-приложения научными сотрудниками, преподавателями и администраторам.

ЗАКЛЮЧЕНИЕ

Во время выполнения квалификационной работы выполнен анализ предметной области, а на основе этого анализа проведено выявление цели, обоснование проблем, основание для выбора технологии реализации и выбранной системы. Выбраны инструменты программного обеспечения и основной язык программирования для разработки проекта.

Спроектирована архитектура автоматизированной информационной системы учета научных трудов преподавателей. Для этого был выбран стек технологий, дизайн системы, база данных и интерфейс, для объединения всего между собой.

При разработке системы созданы и протестированы все компоненты системы. К ним относятся: фреймворки, база данных и интерфейс системы. Также для разработки использована интегрированная среда программирования Microsoft Visual Studio 2019 на платформе ASP.NET Core с использованием шаблона MVC на языке C#, с использованием CSS, HTML и база данных MySQL.

*Я подтверждаю, что настоящая работа написана
лично мной, не нарушает интеллектуальные
права третьих лиц и не содержит сведений,
составляющих государственную тайну*

*Евгф / Егорова Александра Александровна
09.06.2022*

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ И ЛИТЕРАТУРЫ

1. Адам, Фримен ASP.NET Core MVC с примерами на C# для профессионалов. Руководство / Фримен Адам. – М.: Диалектика / Вильямс, 2017. – 483с.
2. Андрей Зиборов «Visual C# 2010», 117–119с.
3. Баженова, И. Ю. Основы проектирования приложений баз данных / Баженова И. Ю. – М.: Национальный Открытый Университет "ИНТУИТ", 2016. – 215 с.
4. Базы данных. Учебник для высших учебных заведений. Под ред. А.Д. Хомоненко. – С-Петербург. – Корона принт.- 2002.
5. Белов, В.В. Проектирование информационных систем: Учебник / В.В. Белов. - М.: Академия, 2018. - 144 с.
6. Бен-Ган, Ицик Microsoft SQL Server 2012. Создание запросов. Учебный курс Microsoft (+ CD-ROM) / Ицик Бен-Ган , Деян Сарка , Рон Талмейдж. - Москва: Высшая школа, 2014. – 720 с.
7. Буч, Грейди Язык UML. Руководство пользователя / Грейди Буч , Джеймс Рамбо , Айвар Джекобсон. - М.: ДМК, 2015. - 432 с. Бишоп, Дж. C# в кратком изложении / Дж. Бишоп, Н. Хорспул. – М.: Бином. Лаборатория знаний, 2013. – 472 с.
8. Ватсон, К. C#. / К. Ватсон, М. Беллиназо, О. Корнс, Д. Эспиноза. – М.: Лори, 2005. – 863 с.
9. Ватсон, К. Visual C# 2010: Полный курс. – М.: Диалектика, 2013. – 356с.
10. Вейскас Д. Эффективная работа с Microsoft SQL Server. – СПб: «Питер Пресс», 2006. – 856 стр.
11. Вирт, Н. Алгоритмы и структуры данных. – СПб: Невский диалект, 2008.

12. Гаевский, А.Ю. 100% самоучитель. Создание Web-страниц и Web-сайтов. HTML и JavaScript / А.Ю. Гаевский, В.А. Романовский. – М.: Триумф, 2014. – 464 с.
13. Гончаров, А. Web-дизайн / А. Гончаров. – Москва: ИЛ, 2016. - 320 с.
14. Горев А., Макашарипов С., Ахьян Р. Эффективная работа с СУБД. – СПб: «Питер», 2004. – 704с.
15. Дакетт, Джон HTML и CSS. Разработка и дизайн веб-сайтов (+ CD-ROM) / Джон Дакетт. - Москва: Наука, 2014. – 480 с.
16. Джеймс, Чамберс ASP.NET Core. Разработка приложений. Руководство / Чамберс Джеймс. – М.: Питер, 2018. – 330 с.
17. Джесс, Чедвик ASP.NET MVC 4. Разработка реальных веб-предложений с помощью ASP.NET MVC / Чедвик Джесс. – М.: Диалектика / Вильямс, 2013. – 854с.
18. Дино, Эспозито Программирование на основе Microsoft ASP.NET MVC / Эспозито Дино. - М.: Русская Редакция, 2012. – 237 с.
19. Дронов, В. HTML 5, CSS 3 и Web 2.0. Разработка современных Web-сайтов / В. Дронов. – М.: БХВ-Петербург, 2014. – 182 с.
20. Зиборов, В.В. Visual C# 2012 на примерах. – СПб: БХВ-Петербург, 2013. – 480 с.
21. Илюшечкин, В. М. Основы использования и проектирования баз данных. Учебник / В.М. Илюшечкин. - М.: Юрайт, 2015. – 214 с.
22. Ищенко, В. А. 100% самоучитель. Web-дизайн. Создавай свои сайты / В.А. Ищенко. - М.: Технолоджи-3000, Триумф, 2016. – 144 с.
23. Йордон, Эдвард Объектно-ориентированный анализ и проектирование систем / Эдвард Йордон , Карл Аргила. – М.: ЛОРИ, 2014. – 264 с.
24. Казанский, А.А. Программирование на Visual C#: учебное пособие для вузов. — М.: Юрайт, 2019. – 192 с.

25. Каратыгин С., Тихонов А., Долголаптев В. Базы данных: простейшие средства обработки информации, электронные таблицы, системы управления базами данных: В 2 т. – М.: АВР, 1995.
26. Кормен, Т. Х. Алгоритмы: построение и анализ. – М.: Вильямс, 2013.
27. Крёнке Д.А. Теория и практика построения баз данных. 9-е изд. – СПб.: Питер, 2006
28. Кудрина, Е. В. Программирование в среде Visual Studio .NET: разработка приложений на языке С#. – Саратов: ООО Издательство "Кубик", 2010.
29. Малыхина, М.П. Базы данных: основы, проектирование, использование: учебное пособие / М. П. Малыхина, – Санкт-Петербург: БХВ-Петербург, 2004, – 512 с.
30. Никсон, Р. Создаем динамические веб-сайты с помощью PHP, MySQL, JavaScript, CSS и HTML5 / Р. Никсон. – М.: Питер, 2015. – 228с.
31. Нортроп, Т. Основы разработки приложений на платформе Microsoft.NET Framework. Учебный курс Microsoft. Перевод с англ./ Т. Нортроп, Ш. Уилдермьюс, Б. Райан. – М.: «Русская редакция», 2007. — 864 с.
32. Петкович, Д. Microsoft SQL Server 2012. Руководство для начинающих / Д. Петкович. – М.: БХВ-Петербург, 2013. – 816 с.
33. Прайс Дж., Гандерлой М. Visual C#.Net Полное руководство. 2004
34. Проектирование баз данных. СУБД Sql Server. Учебное пособие. - М.: Горячая линия – Телеком, 2013. – 240 с.
35. Работа с базами данных на языке С#. Технология ADO.NET. Учебное пособие / О.Н. Евсеева, А.Б. Шамшев – Ульяновск: УлГТУ, 2009 – 170с.
36. Селко, Джо Стилль программирования Джо Селко на SQL / Джо Селко. - М.: Русская Редакция, Питер, 2006. –196 с.

37. Создай свой веб-сайт с помощью HTML и CSS. – М.: Питер, 2013. – 569 с.
38. Чебыкин, Ростислав Самоучитель HTML и CSS. Современные технологии / Ростислав Чебыкин. – Москва: Огни, 2012. – 624 с.
39. Чедвик, Джесс ASP.NET MVC 4. Разработка реальных веб-приложений с помощью ASP.NET MVC / Джесс Чедвик. – М.: Диалектика-Вильямс, 2012. – 989с.
40. Начало работы с ASP.NET Core // Metanit URL: <https://metanit.com/sharp/aspnet5/1.2.php> (дата обращения: 02.03.2022).

Евф / Егорова А.А

Приложение

Фрагменты кода, которые использовались при разработке АИС:

Добавление пользователя преподавателя

```
public IActionResult CreateUser()
{
    List<string> prepodList = new List<string>();
    var prepodss = db.Prepods.ToList();
    var userss = db.Users.ToList();

    foreach (Prepod pr in prepodss)
    {
        prepodList.Add(pr.FIO);
    }
    foreach (Prepod pr in prepodss)
    {
        foreach (User us in userss)
            if (pr.email == us.Email)
            {
                prepodList.Remove(pr.FIO);
            }
    }

    ViewBag.PrepodList = prepodList.ToList();
    ViewBag.Prepods = db.Prepods.ToList();
    //ViewBag.Users = db.Users.ToList();
    return View();
}
```

Вывод списка пользователей:

```
public async Task<IActionResult> SpisokUser(int page = 1)
{
    int pageSize = 5;
    IQueryable<User> source = db.Users.Include(x => x.Role);
    var count = await source.CountAsync();
    var items = await source.Skip((page - 1) *
    pageSize).Take(pageSize).ToListAsync();
    PageViewModel pageViewModel = new PageViewModel(count, page,
    pageSize);
    ОбщаяМодель4 viewModel = new ОбщаяМодель4
    {
        PageViewModel = pageViewModel,
        users= items
    };
    ViewBag.Users = db.Users.ToList();
    return View(viewModel);
}
```

Редактирование научных трудов

```
[Authorize]
public async Task<IActionResult> Edit(int? id)
{
    if (id != null)
    {
        scientific_work user = await
db.scientific_Works.FirstOrDefaultAsync(p => p.id == id);
        if (user != null)
            return View(user);
    }
    return NotFound();
}
[HttpPost]
[Authorize]
public async Task<IActionResult> Edit(scientific_work user)
{
    db.scientific_Works.Update(user);
    await db.SaveChangesAsync();
    //return RedirectToAction("Index");
    return Redirect("~/Home/Index");
}
```

Удаление научных трудов

```
[HttpGet]
[ActionName("Delete")]
[Authorize]
public async Task<IActionResult> ConfirmDelete(int? id)
{
    if (id != null)
    {
        scientific_work user = await
db.scientific_Works.FirstOrDefaultAsync(p => p.id == id);
        if (user != null)
            return View(user);
    }
    return NotFound();
}

[HttpPost]
[Authorize]
public async Task<IActionResult> Delete(int? id)
{
    if (id != null)
    {
        scientific_work user = await
db.scientific_Works.FirstOrDefaultAsync(p => p.id == id);
        if (user != null)
        {
            db.scientific_Works.Remove(user);
        }
    }
}
```

```

        await db.SaveChangesAsync();
        return Redirect("~/Home/Index");
    }
}
return NotFound();
}
}
Сортировка
[Authorize]
public ActionResult Index(int? yer, string name, int fi)
{
    IQueryable<scientific_work> users =
db.scientific_Works.Include(p => p.Prepod);
    if (yer != null && yer != 0)
    {
        users = users.Where(p => p.year == yer);
    }
    if (!String.IsNullOrEmpty(name))
    {
        users = users.Where(p => p.Name.Contains(name));
    }
    if (fi != null && fi != 0)
    {
        users = users.Where(p => p.PrepodId==fi);
    }

    List<scientific_work> companies =
db.scientific_Works.ToList();
    List<Prepod> prepods2 = db.Prepods.ToList();
    // устанавливаем начальный элемент, который позволит выбрать
всех
    companies.Insert(0, new scientific_work { Name = "Все", id =
0 });
    prepods2.Insert(0, new Prepod { FIO = "Все", id = 0 });

    WorkListViewModel viewModel = new WorkListViewModel
    {
        sc_work = users.ToList(),
        year = new SelectList(companies, "year", "year"),
        FIO = new SelectList(preposts2, "id", "FIO"),
        Name = name,
        prepod=db.Prepods.ToList()
    };

    return View(viewModel);
}

```

Экспорт в Excel

[Authorize]

```

public IActionResult ExportToExcel(int? yer, int fi)
{
    // Getting the information from our mimic db
    var users = db.scientific_Works;
    var prepds = db.Prepods;

    // Start exporting to Excel
    var stream = new MemoryStream();
    ExcelPackage.LicenseContext = LicenseContext.NonCommercial;

    using (var xlPackage = new ExcelPackage(stream))
    {
        // Define a worksheet
        var worksheet =
xlPackage.Workbook.Worksheets.Add("НаучныеТруды");

        // Styling
        var customStyle =
xlPackage.Workbook.Styles.CreateNamedStyle("CustomStyle");
        customStyle.Style.Font.UnderLine = true;
        customStyle.Style.Font.Color.SetColor(Color.Red);

        // First row
        var startRow = 5;
        var row = startRow;

        worksheet.Cells["A1"].Value = "Список научных трудов";
        using (var r = worksheet.Cells["A1:E1"])
        {
            r.Merge = true;
            r.Style.Font.Size = 20;
            r.Style.Font.Color.SetColor(Color.Black);
            r.Style.Fill.PatternType =
OfficeOpenXml.Style.ExcelFillStyle.Solid;
            r.Style.Fill.BackgroundColor.SetColor(Color.White);
        }

        worksheet.Cells["A4"].Value = "Название";

        worksheet.Column(2).AutoFit(10f);
        worksheet.Column(1).AutoFit(108f);
        worksheet.Column(3).AutoFit(10f);
        worksheet.Column(4).AutoFit(5f);
        worksheet.Column(5).AutoFit(34f);

        worksheet.Cells["B4"].Value = "Форма";
        worksheet.Cells["C4"].Value = "Год";

        worksheet.Cells["D4"].Value = "Кол-во страниц";
        worksheet.Cells["E4"].Value = "Автор";
    }
}

```

```

        worksheet.Cells["A4:E4"].Style.Fill.PatternType =
OfficeOpenXml.Style.ExcelFillStyle.Solid;

worksheet.Cells["A4:E4"].Style.Fill.BackgroundColor.SetColor(Color.Yellow
);

```

```

row = 5;
foreach (var user in users)
{
    //Вывод всех записей
    if (yer==0 && fi==0)
    {
        worksheet.Cells[row, 1].Value = user.Name;
        worksheet.Cells[row, 2].Value = user.Forma;
        worksheet.Cells[row, 3].Value = user.year;
        worksheet.Cells[row, 4].Value = user.volume;
        foreach (var prep in preps)
        {
            if (user.PrepodId==prep.id)
            {
                worksheet.Cells[row, 5].Value = prep.FIO;
            }
        }
        row++; // row = row + 1;
    }

    //только год
    if (yer != 0 && fi == 0)
    {
        if (user.year==yer)
        {
            worksheet.Cells[row, 1].Value = user.Name;
            worksheet.Cells[row, 2].Value = user.Forma;
            worksheet.Cells[row, 3].Value = user.year;
            worksheet.Cells[row, 4].Value = user.volume;
            foreach (var prep in preps)
            {
                if (user.PrepodId == prep.id)
                {
                    worksheet.Cells[row, 5].Value = prep.FIO;
                }
            }
            row++; // row = row + 1;
        }
    }

    //только фамилия
    if (yer == 0 && fi != 0)
    {
        if (user.PrepodId == fi)

```

```

        {
            worksheet.Cells[row, 1].Value = user.Name;
            worksheet.Cells[row, 2].Value = user.Forma;
            worksheet.Cells[row, 3].Value = user.year;
            worksheet.Cells[row, 4].Value = user.volume;
            foreach (var prep in preps)
            {
                if (user.PrepodId == prep.id)
                {
                    worksheet.Cells[row, 5].Value =
prep.FIO;

                }
            }
            row++; // row = row + 1;
        }
    }

    //год и фамилия
    if (yer != 0 && fi != 0)
    {
        if (user.PrepodId == fi)
        {
            if (user.year == yer)
            {
                worksheet.Cells[row, 1].Value = user.Name;
                worksheet.Cells[row, 2].Value = user.Forma;
                worksheet.Cells[row, 3].Value = user.year;
                worksheet.Cells[row, 4].Value = user.volume;
                foreach (var prep in preps)
                {
                    if (user.PrepodId == prep.id)
                    {
                        worksheet.Cells[row, 5].Value =
prep.FIO;

                    }
                }
                row++; // row = row + 1;
            }
        }
    }

    xlPackage.Workbook.Properties.Title = "Список научных
трудов";
    xlPackage.Workbook.Properties.Author = "Аделина";

    xlPackage.Save();
}

stream.Position = 0;

```

```
        return File(stream, "application/vnd.openxmlformats-officedocument.spreadsheetml.sheet", "НаучныеТруды.xlsx");  
    }
```