

Инженерная школа информационных технологий и робототехники
 Направление подготовки: 09.03.04 «Программная инженерия»
 Отделение информационных технологий

БАКАЛАВРСКАЯ РАБОТА

Тема работы Система мониторинга и контроля научной и учебной исследовательской деятельности студента

УДК 004.65:004.451:004.77:378..14

Студент

Группа	ФИО	Подпись	Дата
8K81	Федоров Илья Владимирович		

Руководитель ВКР

Должность	ФИО	Ученая степень, звание	Подпись	Дата
Старший преподаватель	Мокина Елена Евгеньевна	-		

КОНСУЛЬТАНТЫ ПО РАЗДЕЛАМ:

По разделу «Финансовый менеджмент, ресурсоэффективность и ресурсосбережение»

Должность	ФИО	Ученая степень, звание	Подпись	Дата
Профессор ОСГН	Гасанов Магеррам Али оглы	Д.э.н.		

По разделу «Социальная ответственность»

Должность	ФИО	Ученая степень, звание	Подпись	Дата
Старший преподаватель	Мезенцева Ирина Леонидовна	-		

ДОПУСТИТЬ К ЗАЩИТЕ:

Руководитель ООП	ФИО	Ученая степень, звание	Подпись	Дата
Доцент ОИТ ИШИТР ТПУ	Чердынцев Евгений Сергеевич	К.т.н.		

ПЛАНИРУЕМЫЕ РЕЗУЛЬТАТЫ ОСВОЕНИЯ ООП

Код компетенции	Наименование компетенции
Универсальные компетенции	
У-1	Осуществлять поиск, критический анализ и синтез информации, применять системный подход для решения поставленных задач.
У-2	Определять круг задач в рамках поставленной цели и выбирать оптимальные способы их решения, исходя из действующих правовых норм, имеющихся ресурсов и ограничений.
У-3	Эффективно работать индивидуально и в качестве члена группы, демонстрировать ответственность за результаты работы и готовность следовать корпоративной культуре организации.
У-4	Использовать базовые и специальные знания в области проектного менеджмента для ведения комплексной инженерной деятельности.
У-5	Воспринимать межкультурное разнообразие общества в социально-историческом, этическом и философском контекстах.
У-6	Управлять своим временем, выстраивать и реализовывать траекторию саморазвития на основе принципов образования в течение всей жизни.
У-7	Поддерживать должный уровень физической подготовленности для обеспечения полноценной социальной и профессиональной деятельности.
У-8	Создавать и поддерживать безопасные условия жизнедеятельности, в том числе при возникновении чрезвычайных ситуаций.
Профессиональные компетенции	
П-1	Применять базовые и специальные естественнонаучные и математические знания в области информатики и вычислительной техники, достаточные для комплексной инженерной деятельности.
П-2	Применять базовые и специальные знания в области современных информационных технологий для решения инженерных задач.
П-3	Ставить и решать задачи комплексного анализа, связанные с созданием аппаратно-программных средств информационных и автоматизированных систем, с использованием базовых и специальных знаний, современных аналитических методов и моделей.
П-4	Разрабатывать программные и аппаратные средства (системы, устройства, блоки, программы, базы данных и т.п.) в соответствии с техническим заданием и с использованием средств автоматизации проектирования.
П-5	Проводить теоретические и экспериментальные исследования, включающие поиск и изучение необходимой научно-технической информации, математическое моделирование, проведение эксперимента, анализ и интерпретацию полученных данных, в области создания аппаратных и программных средств информационных и автоматизированных систем.
П-6	Внедрять, эксплуатировать и обслуживать современные программно-аппаратные комплексы, обеспечивать их высокую эффективность, соблюдать правила охраны здоровья, безопасность труда, выполнять требования по защите окружающей среды.

Министерство науки и высшего образования Российской Федерации
 федеральное государственное автономное
 образовательное учреждение высшего образования
 «Национальный исследовательский Томский политехнический университет» (ТПУ)

Инженерная школа информационных технологий и робототехники
 Направление подготовки: 09.03.04 «Программная инженерия»
 Отделение информационных технологий

УТВЕРЖДАЮ:
 Руководитель ООП
 _____ Чердынцев Е.С.
 (Подпись) (Дата)

ЗАДАНИЕ
на выполнение выпускной квалификационной работы

В форме:

Бакалаврской работы

Студенту:

Группа	ФИО
8К81	Федорову Илье Владимировичу

Тема работы:

Система мониторинга и контроля научной и учебной исследовательской деятельности студента	
Утверждена приказом директора (дата, номер)	№ 40-51/с от 09.02.2022

Срок сдачи студентом выполненной работы:	06.06.2022
--	------------

ТЕХНИЧЕСКОЕ ЗАДАНИЕ:

Исходные данные к работе	Объектом проектирования является программное решение, обеспечивающее возможность мониторинга и контроля учебной и научной деятельности студентов в научно-образовательной организации.
---------------------------------	--

Перечень подлежащих исследованию, проектированию и разработке вопросов	1. Обзор предметной области. 2. Проектирование модуля. 3. Программная реализация модуля. 4. Анализ результатов разработки. 5. Финансовый менеджмент, ресурсоэффективность и ресурсосбережение. 6. Социальная ответственность.
Перечень графического материала	1. Fishbone диаграмма. 2. Диаграмма вариантов использования. 3. Диаграмма в нотации IDEF1x. 4. Диаграмма в нотации EPC. 6. Пояснительные скриншоты приложения. 7. Матрица SWOT-анализа. 8. Диаграмма Ганта.
Консультанты по разделам выпускной квалификационной работы	
Раздел	Консультант
Финансовый менеджмент, ресурсоэффективность и ресурсосбережение	Гасанов Магеррам Али оглы
Социальная ответственность	Мезенцева Ирина Леонидовна

Дата выдачи задания на выполнение выпускной квалификационной работы по линейному графику	01.03.2022
---	------------

Задание выдал руководитель / консультант (при наличии):

Должность	ФИО	Ученая степень, звание	Подпись	Дата
Старший преподаватель	Мокина Елена Евгеньевна	-		

Задание принял к исполнению студент:

Группа	ФИО	Подпись	Дата
8K81	Федоров Илья Владимирович		

Министерство науки и высшего образования Российской Федерации
 федеральное государственное автономное
 образовательное учреждение высшего образования
 «Национальный исследовательский Томский политехнический университет» (ТПУ)

Инженерная школа информационных технологий и робототехники

Направление подготовки: 09.03.04 «Программная инженерия»

Уровень образования: бакалавр

Отделение информационных технологий

Период выполнения: весенний семестр 2021-2022 года

Форма представления работы:

Бакалаврская работа

**КАЛЕНДАРНЫЙ РЕЙТИНГ-ПЛАН
выполнения выпускной квалификационной работы**

Срок сдачи студентом выполненной работы:	06.06.2022
--	------------

Дата контроля	Название раздела (модуля) / вид работы (исследования)	Максимальный балл раздела (модуля)
	Анализ предметной области	20
	Проектирование модуля	20
	Программная реализация модуля	20
	Финансовый менеджмент, ресурсоэффективность и ресурсосбережение	20
	Социальная ответственность	20

СОСТАВИЛ:

Руководитель ВКР

Должность	ФИО	Ученая степень, звание	Подпись	Дата
Старший преподаватель	Мокина Елена Евгеньевна	-		

СОГЛАСОВАНО:

Руководитель ООП

Должность	ФИО	Ученая степень, звание	Подпись	Дата
Доцент ОИТ ИШИТР ТПУ	Чердынцев Евгений Сергеевич	К.т.н.		

ЗАДАНИЕ ДЛЯ РАЗДЕЛА «ФИНАНСОВЫЙ МЕНЕДЖМЕНТ, РЕСУРСООБЪЕКТИВНОСТЬ И РЕСУРСОСБЕРЕЖЕНИЕ»

Студенту:

Группа	ФИО
8К81	Федорову Илье Владимировичу

Школа	ИШИТР	Отделение школы (НОЦ)	ОИТ
Уровень образования	Бакалавриат	Направление	09.03.04. «Программная инженерия»

Исходные данные к разделу «Финансовый менеджмент, ресурсоэффективность и ресурсосбережение»:

1. <i>Стоимость ресурсов научного исследования (НИ): материально-технических, энергетических, финансовых, информационных и человеческих</i>	Оклад руководителя – 30000 руб. Оклад разработчика – 15000 руб.
2. <i>Нормы и нормативы расходования ресурсов</i>	Премимальный коэффициент 30%; Доплаты и надбавки руководителя 40%; Доплаты и надбавки разработчика 20%; Дополнительной заработной платы 15%; Накладные расходы 15%; Районный коэффициент 1,3.
3. <i>Используемая система налогообложения, ставки налогов, отчислений, дисконтирования и кредитования</i>	Тариф отчислений во внебюджетные фонды 7,6%

Перечень вопросов, подлежащих исследованию, проектированию и разработке:

1. <i>Оценка коммерческого потенциала, перспективности и альтернатив проведения НИ с позиции ресурсоэффективности и ресурсосбережения</i>	Определение потенциального потребителя результатов исследования. Анализ конкурентных технических решений. SWOT-анализ разработанной стратегии.
2. <i>Планирование и формирование бюджета научных исследований</i>	Определение структуры работы. Расчет трудоемкости выполнения работ. Подсчет бюджета исследования
3. <i>Определение ресурсной (ресурсосберегающей), финансовой, бюджетной, социальной и экономической эффективности исследования</i>	Рассчитать показатели финансовой эффективности, ресурсоэффективности и эффективности исполнения

Перечень графического материала:

1. Оценка конкурентоспособности технических решений
2. Матрица SWOT
3. График проведения и бюджет НИ
4. Оценка ресурсной, финансовой и экономической эффективности НИ

Дата выдачи задания для раздела по линейному графику	11.02.2022
--	------------

Задание выдал консультант:

Должность	ФИО	Ученая степень, звание	Подпись	Дата
Профессор ОСГН	Гасанов Магеррам Али оглы	д.э.н.		04.03.2022

Задание принял к исполнению студент:

Группа	ФИО	Подпись	Дата
8К81	Федоров Илья Владимирович		04.03.2022

ЗАДАНИЕ ДЛЯ РАЗДЕЛА «СОЦИАЛЬНАЯ ОТВЕТСТВЕННОСТЬ»

Студенту:

Группа 8K81		ФИО Федорову Илье Владимировичу	
Школа	Инженерная школа информационных технологий и робототехники	Отделение (НОЦ)	Отделение информационных технологий
Уровень образования	Бакалавриат	Направление/специальность	09.04.04 Программная инженерия

Тема ВКР:

Система мониторинга и контроля научной и учебной исследовательской деятельности студента	
Исходные данные к разделу «Социальная ответственность»:	
<p>Введение</p> <ul style="list-style-type: none"> - Характеристика объекта исследования (вещество, материал, прибор, алгоритм, методика) и области его применения. - Описание рабочей зоны (рабочего места) при разработке проектного решения 	<p><i>Объект исследования:</i> система мониторинга и контроля научной и учебной исследовательской деятельности студента, разработка серверной части с помощью фреймворка Spring Boot на языке программирования Java.</p> <p><i>Область применения:</i> предоставление услуг онлайн.</p> <p><i>Рабочая зона:</i> <u>офис</u>.</p> <p><i>Размеры помещения:</i> 6*4 м.</p> <p><i>Количество и наименование оборудования рабочей зоны:</i> ноутбук.</p> <p><i>Рабочие процессы, связанные с объектом исследования, осуществляемые в рабочей зоне:</i> проектирование, разработка программного обеспечения.</p>
Перечень вопросов, подлежащих исследованию, проектированию и разработке:	
<p>1. Правовые и организационные вопросы обеспечения безопасности при разработке проектного решения:</p> <ul style="list-style-type: none"> - специальные (характерные при эксплуатации объекта исследования, проектируемой рабочей зоны) правовые нормы трудового законодательства; - организационные мероприятия при компоновке рабочей зоны. 	<ul style="list-style-type: none"> - Рациональная организация труда в течение рабочего времени предусмотрена Трудовым Кодексом РФ ФЗ-197 - Рабочее место при выполнении работ сидя регулируется ГОСТом 12.2.032-78
<p>2. Производственная безопасность при разработке проектного решения:</p> <ul style="list-style-type: none"> - Анализ выявленных вредных и опасных производственных факторов 	<p>Вредные факторы:</p> <ul style="list-style-type: none"> - Отклонение показателей микроклимата - Недостаточная освещенность рабочей зоны - Монотонность труда <p>Опасные факторы:</p> <ul style="list-style-type: none"> - Опасность поражения электрическим током <p>Требуемые средства индивидуальной защиты:</p> <ul style="list-style-type: none"> - Коврик - Подставка
<p>3. Экологическая безопасность при разработке проектного решения</p>	<p>Воздействие на селитебную зону: неправильная утилизация компьютерной и оргтехники.</p> <p>Воздействие на литосферу: выброс батареек и других элементов техники, содержащих вредные химические вещества, выброс деталей, сделанных из пластика.</p> <p>Воздействие на гидросферу: попадание токсичных веществ в воду, содержащиеся в деталях компьютера.</p>

	Воздействие на атмосферу: выброс вредных веществ при сжигании деталей ЭВМ.
4. Безопасность в чрезвычайных ситуациях при разработке проектного решения	<p>Возможные ЧС: Природные катастрофы (наводнения, цунами, ураган и т.д.); Геологические воздействия (землетрясения, оползни, обвалы, провалы территории); Техногенные аварии (пожар, сбой в электроснабжении)</p> <p>Наиболее типичная ЧС (пожар)</p>
Дата выдачи задания для раздела по линейному графику	

Задание выдал консультант:

Должность	ФИО	Ученая степень, звание	Подпись	Дата
Старший преподаватель	Мезенцева Ирина Леонидовна			

Задание принял к исполнению студент:

Группа	ФИО	Подпись	Дата
8К81	Федоров Илья Владимирович		

Реферат

Выпускная квалификационная работа выполнена на 116 страницах, содержит 17 рисунков, 24 таблицы, 22 источника литературы.

Ключевые слова: проектирование и разработка программного обеспечения, информационная система, веб-приложение, исследовательская деятельность, мониторинг.

Объектом исследования является научная и учебная исследовательская деятельность в научно-образовательной организации. Предмет исследования – программное решение, обеспечивающее информационную поддержку процессов мониторинга и контроля исследовательской деятельности.

Цель работы – проектирование и разработка программного решения для обеспечения педагогов и руководства вуза удобством понимания и визуализации ситуации проектной деятельности как отдельного студента, так и руководителей, включая ответственного за исследовательскую деятельность в вузе.

В процессе работы была спроектирована и разработана информационная система в соответствии с выявленными требованиями.

Область применения – мониторинг и управление публикационной деятельностью научно-образовательной организации.

В будущем планируется расширение набора данных, предоставляемых системой.

Термины и сокращения

СУБД (Система Управления Базами Данных) – совокупность программных и лингвистических средств общего или специального назначения, обеспечивающих управление созданием и использованием баз данных.

ЕPC (событийная цепочка процессов) — тип блок-схемы, используемой для бизнес-моделирования.

IDEF1x – нотация и метод для разработки реляционных баз данных.

HTTP-сервер – сервер, принимающий HTTP-запросы от клиентов, обычно веб-браузеров, и выдающий им HTTP-ответы, как правило, вместе с HTML-страницей, изображением, файлом или другими данными.

Фреймворк – программное обеспечение, облегчающее разработку и объединение разных компонентов информационной системы.

Spring Framework (Spring) – универсальный фреймворк с открытым исходным кодом для Java-платформы.

MVC (Model-View-Controller) – шаблон проектирования веб-приложений, схема разделения данных приложения и управляющей логики на три отдельных компонента: модель, представление и контроллер — таким образом, что модификация каждого компонента может осуществляться независимо.

ORM (Object–relational mapping) – технология программирования, которая связывает базы данных с концепциями объектно-ориентированных языков программирования, создавая «виртуальную базу данных», состоящую из объектов.

JPQL (Java Persistence Query Language) – объектно-ориентированный язык запросов к реляционным базам данным.

Оглавление

Реферат	9
Термины и сокращения	10
Оглавление	11
Введение	15
1. Описание предметной области	16
1.1. Общие сведения о ведении проектной деятельности в вузе	16
1.1.1. Определение темы творческого или научно-исследовательского проекта	16
1.1.2. Разработка проекта в университете	17
1.2. Постановка проблемы	18
1.3. Требования к разрабатываемой системе	19
1.4. Вывод по разделу	21
2. Проектирование информационной системы	22
2.1. Роли и функциональные возможности разрабатываемой системы	22
2.2. Логическая модель данных	25
2.3. Описание процесса научной деятельности студента с использованием диаграммы EPC	27
2.4. Вывод по разделу	30
3. Программная реализация системы	31
3.1. Обоснование используемых технологий	31
3.1.1. Язык программирования серверной части	31
3.1.2. Система управления базами данных	31
3.1.3. Среда разработки	32
	11

3.2.	Разработка модуля «Авторизация»	32
3.3.	Основной функционал системы	33
3.3.1.	Главная страница	33
3.3.2.	Страница «Преподаватели»	34
3.3.3.	Страница «Студенты»	35
3.3.4.	Страница «Все работы»	36
3.3.5.	Страница профиля	38
3.3.6.	Страница «Отзывы»	39
3.4.	Вывод по разделу	40
4.	Разработка серверной части разрабатываемой системы	41
4.1.	Описание технологии разработки веб-приложения	41
4.2.	Разработка веб-приложения на Spring Boot	42
4.2.1.	Подключение к базе данных MySQL	42
4.2.2.	Работа Hibernate при обмене данными	42
4.2.3.	Создание условий для работы с ORM с использованием JPA	46
4.2.4.	Подключение Spring Security для управления доступом к данным	47
4.2.5.	Использование MVC при разделении компонентов программы	47
4.3.	Вывод по разделу	49
5.	Финансовый менеджмент, ресурсоэффективность и ресурсосбережение	50
5.1.	Оценка коммерческого потенциала и перспективности научных исследований	51
5.1.1.	Потенциальные потребители результатов исследования	51
5.1.2.	Анализ конкурентных технических решений	52
5.1.3.	Технология QuaD	54

5.1.4. SWOT-анализ	55
5.2. Определение возможных альтернатив проведения научных исследований	58
5.3. Планирование работ по научно-техническому исследованию	60
5.3.1. Структура работ в рамках научного исследования	60
5.3.2. Определение трудоемкости выполнения работ	62
5.3.3. Разработка графика проведения научного исследования	63
5.4. Бюджет научно-технического исследования (НТИ)	69
5.4.1. Расчет материальных затрат	69
5.4.2. Расчет затрат на специальное оборудование для научных работ	70
5.4.3. Основная заработная плата исполнителя темы	71
5.4.4. Расчет дополнительной заработной платы	74
5.4.5. Отчисления во внебюджетные фонды	75
5.4.6. Контрагентные расходы	76
5.4.7. Накладные расходы	76
5.4.8. Формирование бюджета затрат научно-исследовательского проекта	77
5.5. Определение ресурсной (ресурсосберегающей), финансовой, бюджетной, социальной и экономической эффективности исследования	77
5.6. Вывод по главе	80
6. Социальная ответственность	81
6.1. Введение	81
6.2. Правовые и организационные вопросы обеспечения безопасности	81
6.2.1. Правовые нормы трудового законодательства	81

6.2.2. Эргономические требования к правильному расположению и компоновке рабочей зоны	82
6.3. Производственная безопасность	83
6.3.1. Отклонение показателей микроклимата	84
6.3.2. Недостаточная освещенность рабочей зоны	85
6.3.3. Монотонность труда	86
6.3.4. Опасность поражения электрическим током	86
6.4. Экологическая безопасность	88
6.5. Безопасность в чрезвычайных ситуациях	89
6.6. Выводы по разделу	90
Заключение	91
Список источников	92
Приложение А. Код DDL базы данных	95
Приложение Б. Настройка источника данных и JPA	100
Приложение В. Реализация JPA репозитория для сущности Student	101
Приложение Г. Настройка Entity классов для таблиц Teacher, Science_specialization и Teacher_specialization	102
Приложение Д. Настройка доступа к ресурсам веб-приложения	105
Приложение Е. Код Контроллера для страницы «Все работы»	105
Приложение Ж. Код Представления для страницы «Работы»	108
Приложение И. Hibernate запросы при загрузке страницы «Работы»	111
Приложение К. Ссылка на репозиторий Github с исходным кодом	116

Введение

Университет – это место не только для прохождения основной учебной программы. Наряду с освоением специальности, студенты раскрывают свой творческий и интеллектуальный потенциал и пробуют себя в исследовательской деятельности.

Учебное и научное исследование направлено на выработку у студента определенных знаний и навыков, оттачивает как его самостоятельность, так и soft-skills при работе с его руководителем.

Выбор руководителя исследовательского проекта имеет огромное значение для эффективной работы по исследовательскому проекту. От него зависят дальнейшие успехи студента.

В ходе данного проекта будет раскрыта суть приложения для отслеживания работ и тем исследовательской деятельности каждого студента вуза и руководителей.

Главной целью создания описанной системы выступает обеспечение педагогов и руководства вуза удобством понимания и визуализации ситуации проектной деятельности как отдельного студента, так и руководителей, включая ответственного за исследовательскую деятельность в вузе.

Тема актуальна благодаря пользе приложения как средства, порождающего правила взаимоотношений между преподавателями и студентами университета или влияющего на них. Роль такой системы будет определяться улучшением качественных признаков, которые воздействуют на будущих выпускников со стороны учебного заведения на протяжении всего времени их обучения.

Главными чертами программы являются интегрируемость и востребованность.

1. Описание предметной области

1.1. Общие сведения о ведении проектной деятельности в вузе

1.1.1. Определение темы творческого или научно-исследовательского проекта

Начиная с первого курса всем студентам предлагается испытать себя в проектной деятельности в рамках творческого проекта и научно-исследовательской деятельности на протяжении всего процесса обучения. Студентам предлагается выбрать тему исследования, которую можно менять от семестра к семестру, что является весьма трудным и ответственным этапом, в особенности для студентов 1 и 2 курса.

Тема должна быть не только интересна на данный момент, но и быть востребована в будущем; она должна вписываться в программу профильного обучения. Тема также должна быть реализуема в имеющихся условиях. По ней можно найти достаточно источников информации, требуемое оборудование и условия для ведения исследования.

Конечно же, было бы хорошо, если бы выбор темы был связан с интересом к ней и научного руководителя, тогда это будут отношения сотрудничества между обучающимися и научным консультантом проекта и исследования.

Важно с самого начала правильно сформулировать тему. Ведь тема – это своего рода визитная карточка исследования, хотя в последствии формулировка будет неоднократно корректироваться. Тема исследования в проблемной форме отразит отношение предмета и объекта исследования, которое предполагается изучить.

1.1.2. Разработка проекта в университете

Предложения извне, личные знакомства или прямое назначение – все это является самостоятельными факторами выбора руководителя из числа работников университета.

Неудобством, стоящим на пути большого количества студентов, является как недостаточная осведомленность, так и малый опыт, а также неуверенность если не в своих силах, то, например, в своей теме. Вузы предоставляют возможность смены научного руководителя, однако в приоритет мы ставим «точное попадание» с первого раза с целью экономии времени и нервов.

Работа под присмотром руководителя – не череда прописанных им действий. Расскажи, что делаешь, и получи совет; не знаешь, что делать – получи тему; не понимаешь, как делать – перейми опыт, посмотри на личном примере. Так в идеале работает сотрудничество студента и преподавателя на исследовательском уровне.

Система мониторинга и контроля исследовательской деятельности студента поможет улучшить поиски такого сотрудничества на университетском уровне, не говоря уже о поддержке студентов, оказавшихся в трудной ситуации – от выбора руководителя до результатов проектной работы, противоположных ожиданиям.

Действительно, именно неоправданные ожидания заставляют обучающегося потерять мотивацию к обучению. Зачастую это происходит на достаточно продолжительный срок, что может говорить как о возможной скорой смене направления или даже учебного заведения, так и о повышении скепсиса к обучению в целом.

1.2. Постановка проблемы

Основная проблема состоит в несвоевременном выборе темы и взаимодействии с руководителем проекта, что тормозит исследование и развитие в выбранном студентом направлении. Можно выделить следующие причины, по которым проблема имеет место быть:

- Слабое информирование. Неудобством, стоящим на пути большого количества студентов, является недостаточная осведомленность о темах, возможных руководителях и этапах выполнения творческих и исследовательских проектов.
- Вынужденная смена руководителя. Неоправданные ожидания в сторону наставника, что влечет за собой поиск нового, заставляют обучающегося потерять мотивацию к обучению. Зачастую это происходит на достаточно продолжительный срок, что может говорить как о возможной скорой смене направления или даже учебного заведения, так и о повышении скепсиса к обучению в целом.

Потеря времени также вытекает из вышеописанной проблемы и является ее неизбежным следствием. На протяжении обучения в университете у студентов должен формироваться ряд очень важных качеств (трудолюбие, тайм-менеджмент, самостоятельность, soft-skills), влияющих на будущие компетенции.

Разработка системы, которая сможет помочь с мониторингом и контролем научной и учебной исследовательской деятельности студента, и ее внедрение позволят минимизировать основную проблему.

На рисунке 1 приведена диаграмма Исикавы, построенная в соответствии с проблемой взаимодействия студентов со своими научными руководителями.



Рисунок 1 – Диаграмма Исикавы для основной проблемы

1.3. Требования к разрабатываемой системе

В итоге ознакомления и анализа предметной области были определены требования к разрабатываемой системе:

1. Возможность авторизации с использованием корпоративной почты Томского политехнического университета.
2. Отображение списка студентов.
3. Фильтрация списка студентов.
4. Отображение списка преподавателей.
5. Фильтрация списка преподавателей.
6. Отображение списка проектов.
7. Фильтрация списка проектов.
8. Отображение информации о студенте:
 - ФИО;
 - группа;
 - школа;

- e-mail;
- список прикрепленных студентов;
- список текущих проектов;
- список завершенных проектов;
- средний балл за завершённые проекты.

9. Отображение информации о преподавателе:

- ФИО;
- отделение;
- ученая степень;
- должность в вузе;
- список компетенций (научных направлений);
- e-mail;
- список текущих проектов;
- список завершённых проектов.

10. Отображение информации о проекте:

- тема;
- студенты-исполнители;
- статус;
- научный руководитель.

11. Фильтрация списка текущих проектов по типу проекта и номеру курса.

12. Возможность редактирования статуса проекта преподавателем:

- создание комментария;
- выставление оценки.

13. Возможность добавления отзыва о студенте.

14. Возможность добавления отзыва о преподавателе.

15. Отображение списка отзывов.

16. Генерация текста электронного письма преподавателю с просьбой дать студенту тему работы.

17. Возможность создания проекта преподавателем.

1.4. Вывод по разделу

В разделе анализа предметной области были представлены общие сведения о ведении проектной деятельности в вузе, а также сформулирована основная проблема, возникающая у студентов и их научных руководителей.

Была предложена разработка информационной «Системы мониторинга и контроля научной и учебной исследовательской деятельности студента» для решения поставленной проблемы. Были выявлены требования к разрабатываемой системе.

2. Проектирование информационной системы

2.1. Роли и функциональные возможности разрабатываемой системы

После составления требований к информационной системе были определены основные роли для построения диаграмм вариантов использования: для авторизованного студента, авторизованного преподавателя, неавторизованного пользователя и для администратора. Неавторизованным пользователям доступен не весь функционал, в то время как авторизованным предоставляется возможность полноценного пользования информационной системой.

Неавторизованный пользователь может авторизоваться в системе, просматривать информацию о студентах, преподавателях и проектах.

На рисунке 2 отображена диаграмма вариантов использования системы неавторизованным пользователем.

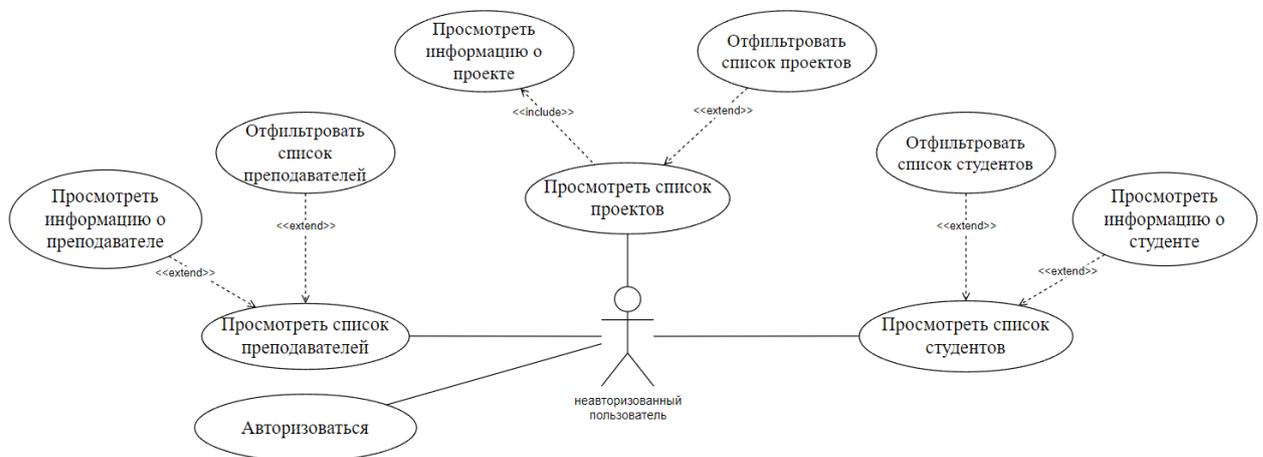


Рисунок 2 – Функциональные возможности неавторизованного пользователя

Авторизованный пользователь может быть студентом и преподавателем.

Помимо основной функциональности информационной системы, доступной неавторизованному пользователю, авторизованному студенту

доступна функция просмотра автоматически сгенерированного текста электронного письма преподавателю с просьбой взять у него тему для последующей работы над ней.

Авторизованный преподаватель может редактировать статус проекта, если работа над ним проходит под его руководством, и создавать новые проекты после подтверждения темы.

Будучи авторизованным, можно добавлять отзывы о конкретных студентах или преподавателях и просматривать список всех отзывов, а также выйти из системы.

На рисунке 3 отображена диаграмма вариантов использования системы авторизованным студентом.

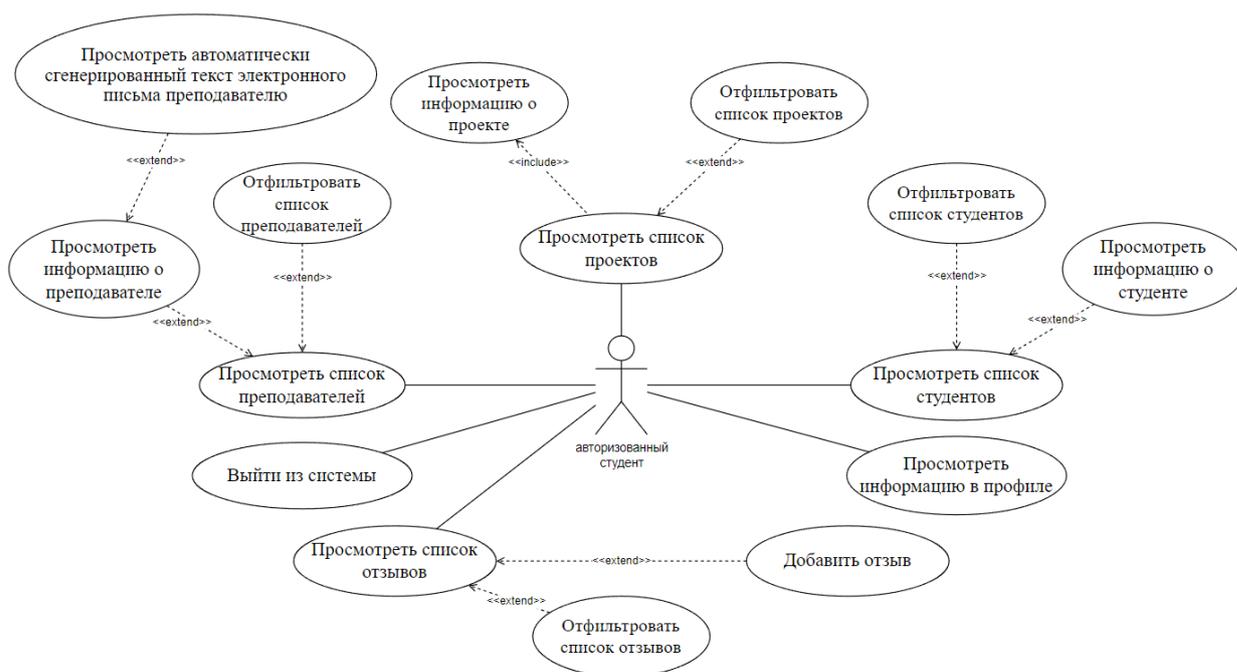


Рисунок 3 – Функциональные возможности авторизованного студента

На рисунке 4 отображена диаграмма вариантов использования системы авторизованным преподавателем.

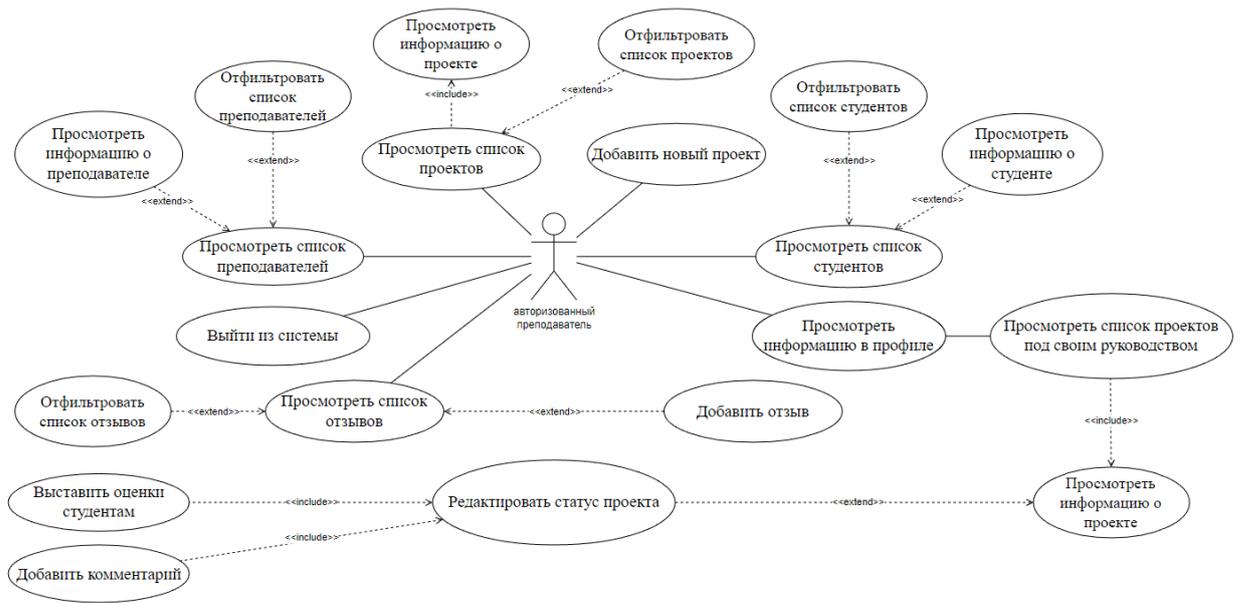


Рисунок 4 – Функциональные возможности авторизованного преподавателя

Администратору доступен функционал, присущий авторизованному преподавателю, а также возможность создавать пользователей.

На рисунке 5 отображена диаграмма вариантов использования системы авторизованным преподавателем.

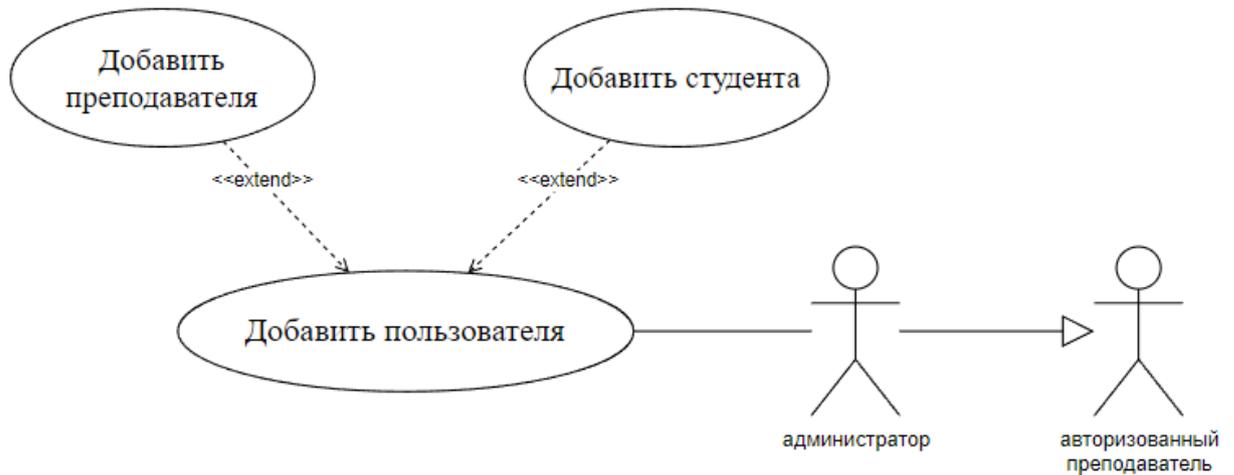


Рисунок 5 – Функциональные возможности администратора

2.2. Логическая модель данных

Логическая модель отображает данные предметной области в виде совокупности информационных объектов и связей между ними. Логическая модель данных является прототипом будущей базы данных и была создана на данном этапе без привязки к конкретной СУБД.

Логическая модель, визуализированная с использованием графической нотации IDEF1x, отображена на рисунке 6.

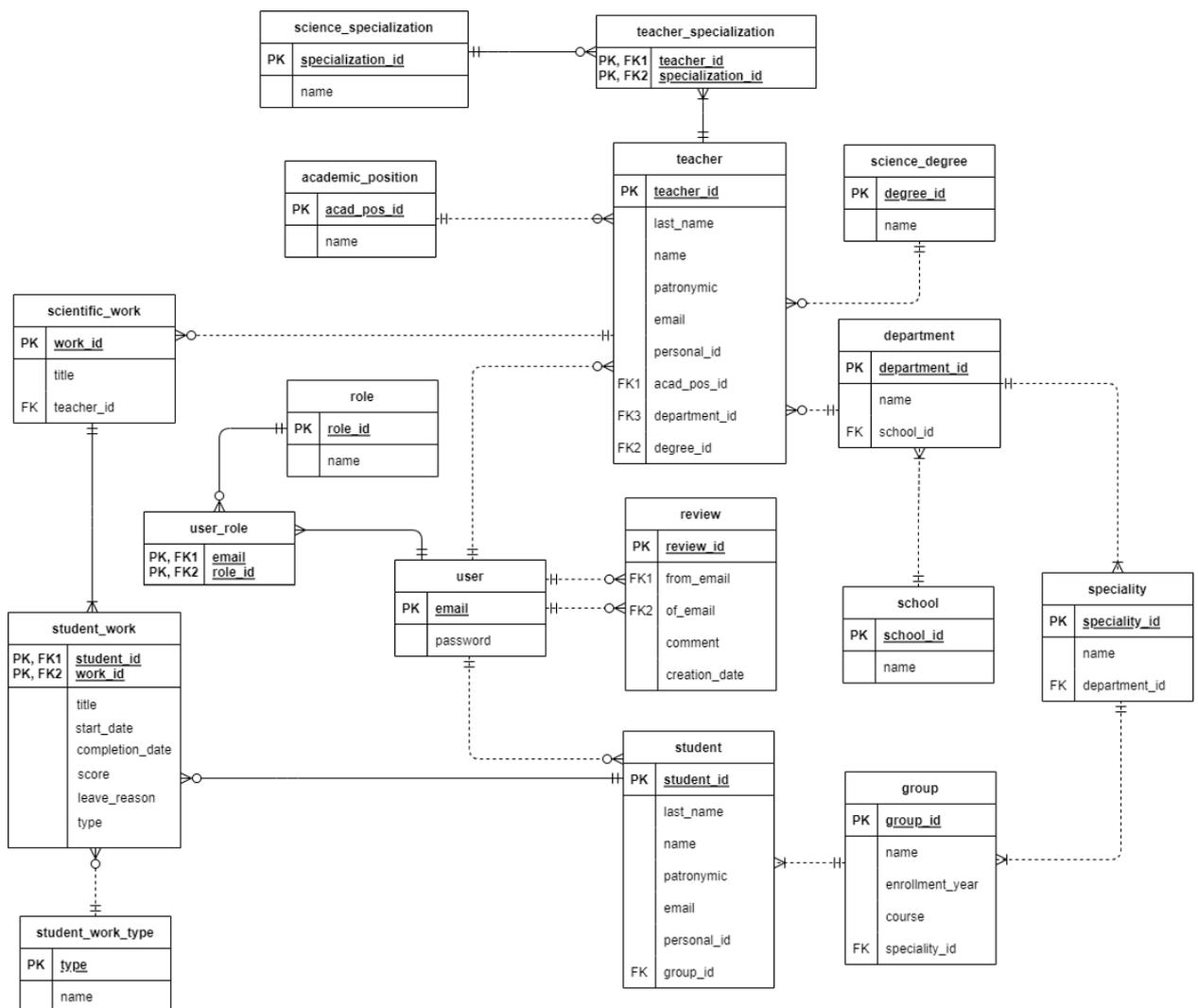


Рисунок 6 – Логическая модель данных

В разрабатываемой системе выделено 17 сущностей:

1. Студент – сущность студента университета. Содержит в себе ФИО студента, e-mail, персональный номер и ссылку на группу.
2. Преподаватель – сущность преподавателя университета. Содержит в себе ФИО преподавателя, e-mail, персональный номер, ссылки на должность в университете, ученую степень и отделение.
3. Группа – сущность группы, состоящей из студентов. Содержит в себе название группы, год поступления студентов группы, номер курса и ссылку на специальность.
4. Специальность – сущность специальности, состоящей из групп. Содержит в себе наименование специальности и ссылку на отделение.
5. Отделение – сущность отделения, состоящего из групп. Содержит в себе наименование специальности и ссылку на школу (факультет).
6. Школа – сущность школы (факультета), состоящей из отделений. Содержит в себе наименование школы.
7. Ученая степень – таблица-словарь ученых степеней. Содержит названия ученых степеней.
8. Академическая должность – таблица-словарь должностей в университете. Содержит названия должностей.
9. Специализация преподавателя – сущность, связывающая преподавателей и научные направления. Содержит в себе ссылки на преподавателя и специализацию.
10. Научная специализация – сущность научных специализаций (направлений). Содержит наименование научного направления.
11. Работа студента – сущность студенческих проектов. Содержит в себе ссылку на научную работу, ссылку на студента, выполняющего конкретную работу, тему работы, дату начала, дату окончания работы, оценку, причину, по которой студент прекратил работу над проектом, тип работы.

12. Тип работы студента – таблица-словарь для типов работ студента. Содержит в себе наименование типа работы, например, научно-исследовательский проект или творческий проект.
13. Научная работа – сущность, связывающая работу студента и преподавателя. Содержит в себе тему работы и ссылку на научного руководителя (преподавателя), под руководством которого проходит или проходила конкретная работа.
14. Пользователь – сущность пользователя, зарегистрированного в информационной системе. Содержит в себе логин (e-mail) и пароль пользователя.
15. Роль – сущность роли пользователя системы. Содержит в себе название роли.
16. Роль пользователя – сущность для объединения роли и пользователя. У пользователя может быть больше одной роли. Сущность содержит в себе ссылки на пользователей и роли.
17. Отзыв – сущность отзывов на студентов и преподавателей. Содержит ссылки на пользователя-автора отзыва, на пользователя, о котором написан отзыв, комментарий – тело отзыва, дату создания отзыва.

Код DDL для базы данных представлен в Приложении А.

2.3. Описание процесса научной деятельности студента с использованием диаграммы EPC

На данном этапе был смоделирован процесс научной деятельности студента в течение семестра в университете с использованием нотации EPC.

Диаграмма процесса научной деятельности студента в течение семестра в университете отображена на рисунках 7-9. Диаграмма демонстрирует процесс от состояния «Нужно сделать и успешно защитить творческий проект или научно-исследовательскую работу» до состояния «Проектная работа в семестре завершена».

Часть диаграммы EPC на рисунке 7 показывает, с чего начинается основной процесс научной деятельности студента и как именно происходит подготовка к работе в течение семестра.

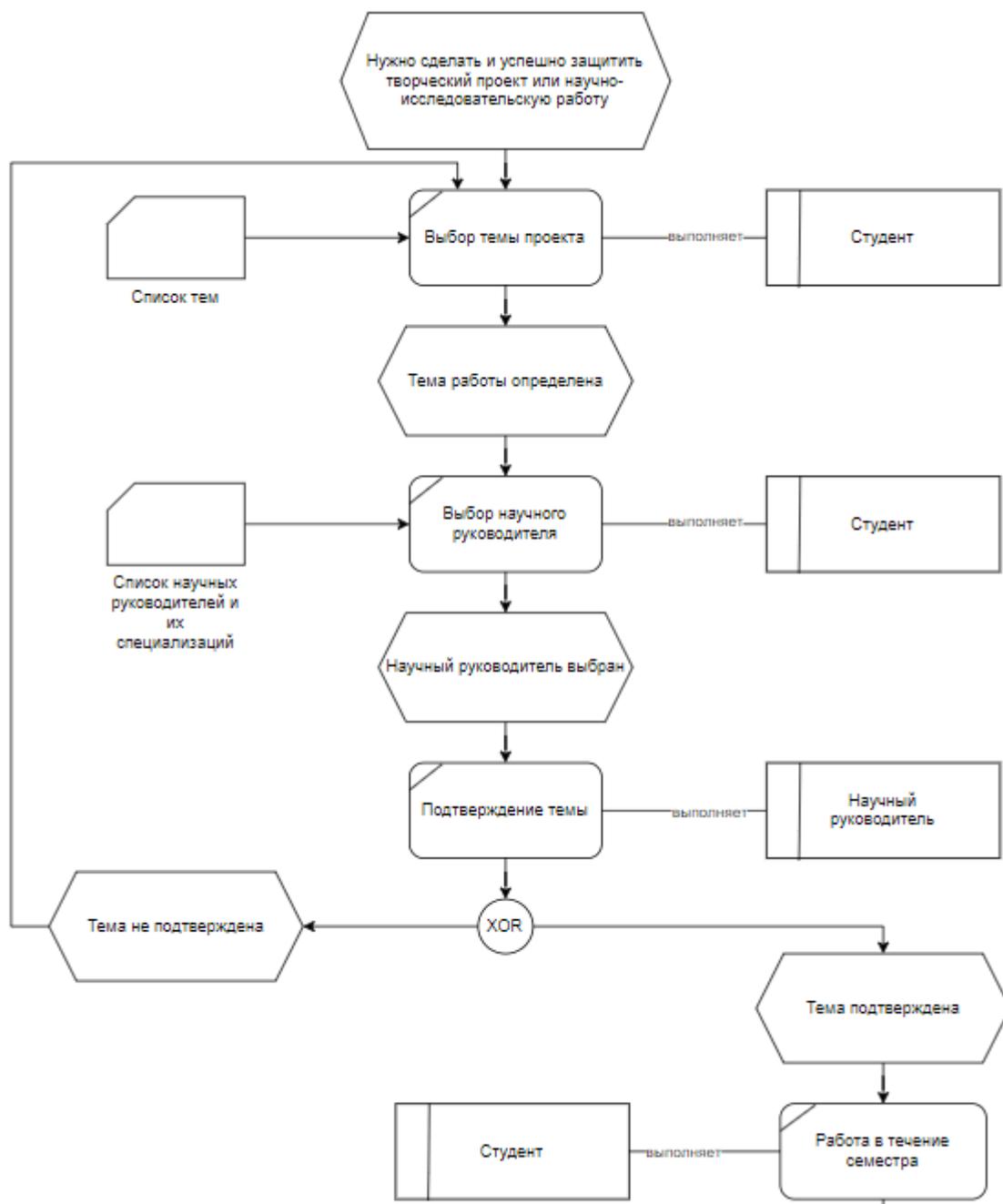


Рисунок 7 – Процесс научной деятельности студента до состояния «Работать в течение семестра»

Часть диаграммы на рисунке 8 показывает связь событий и функций, начиная от состояния «Работать в течение семестра» и заканчивая состоянием «Защита исследовательской работы».

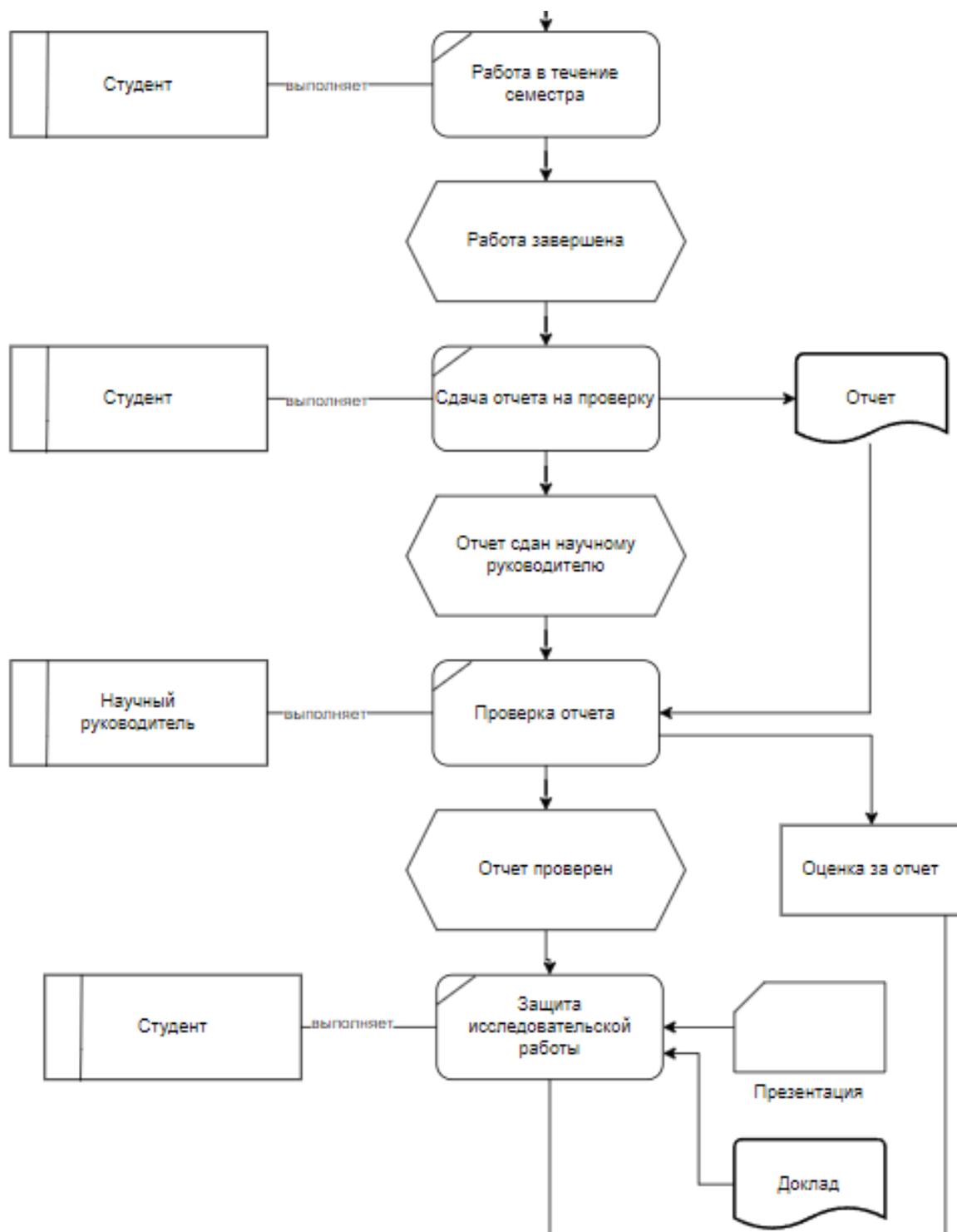


Рисунок 8 – Процесс научной деятельности студента до состояния «Защита исследовательской работы»

Рисунок 9 демонстрирует завершающие этапы диаграммы. ЕРС в целом выстроена по принципу иерархического упорядочивания.

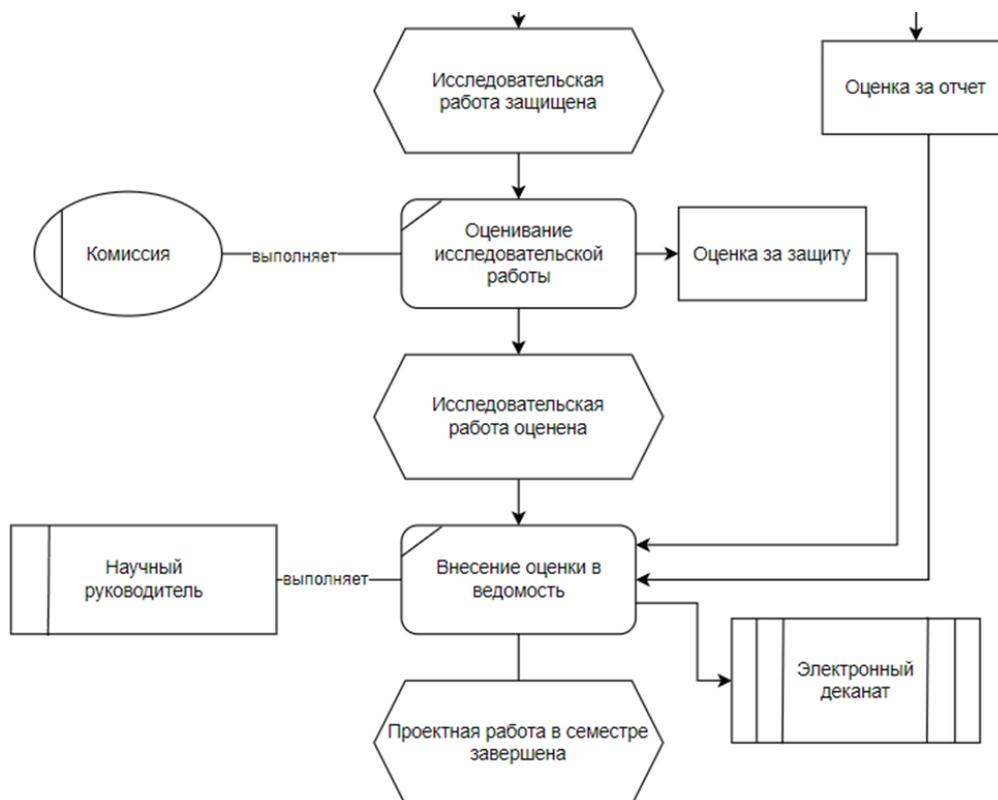


Рисунок 9 – Процесс научной деятельности студента до состояния «Проектная работа в семестре завершена»

В разрабатываемой информационной системе члены комиссии, принимающей защиту работы студента, не являются участниками процесса научной деятельности студента.

2.4. Вывод по разделу

В данном разделе были выделены роли пользователей приложения и их функциональные возможности, описаны бизнес-процессы предметной области.

Этапы проектирования были визуализированы с помощью таких средств, как диаграмма вариантов использования, логическая модель данных в нотации IDEF1x и диаграмма бизнес-процессов в нотации ЕРС.

3. Программная реализация системы

3.1. Обоснование используемых технологий

Перед непосредственной разработкой информационной системы необходимо определиться со стеком используемых технологий.

3.1.1. Язык программирования серверной части

Объективность выбора языка программирования для разработки информационной системы особенно важна в современном мире.

На принятие решение повлияли следующие критерии:

1. Опыт работы с языком.
2. Тип разрабатываемого приложения.
3. Степень поддержки сообщества.

Для проекта большего размера критериев будет гораздо больше, однако для разрабатываемой системы достаточно предложенных.

Было принято решение о том, что информационная система будет разрабатываться в виде веб-приложения из соображений удобства пользования ею. Самые популярные языки программирования для разработки бэкенда веб-приложений – это Java и C#.

Из C# и Java был выбран Java с преимуществом в опыте разработки и масштабах Java-сообщества на просторах Интернета.

3.1.2. Система управления базами данных

Самой распространенной системой управления базами данных, используемой веб-приложениями, написанными на Java, является PostgreSQL. Еще одной свободно распространяемой и не менее популярной системой управления базами данных является MySQL. MySQL используется такими компаниями, как Github, Google, PayPal.

Критериями выбора СУБД выступали простота использования и возможность масштабирования.

MySQL характеризуется высокой масштабируемостью и гибкостью для всех типов данных и пользователей. Он предлагает высокую производительность и доступность для использования в любое время и из любого места [1].

Более того, в Томском политехническом университете разработана система электронного хранилища на основе СУБД Oracle [2]. В нынешних реалиях разработка с использованием СУБД Oracle стала немного сложнее. СУБД MySQL максимально совместима с СУБД Oracle, и в качестве СУБД разрабатываемого приложения была выбрана именно СУБД MySQL.

3.1.3. Среда разработки

Самыми популярными средами разработки на Java являются Eclipse, NetBeans и IntelliJ IDEA.

Для данного проекта была выбрана IntelliJ IDEA в связи с опытом работы на ней и наличием лицензионной версии, позволяющей пользоваться большей функциональностью популярной среды разработки от JetBrains.

3.2. Разработка модуля «Авторизация»

Для работы с веб-приложением пользователю не обязательно быть авторизованным, если его целью является просмотр систематизированных данных. Однако полная функциональность системы доступна после авторизации с использованием корпоративной почты Томского политехнического университета.

На рисунке 10 продемонстрирована форма авторизации пользователя. Она совпадает с формой авторизации Корпоративного портала ТПУ [3].

Вход в систему учета проектной деятельности ТПУ

Введите имя пользователя и пароль для регистрации с уникальным паролем

Имя польз.
Пароль
Тип

Несанкционированное использование этого Web-узла запрещено и преследуется в соответствии с гражданским или уголовным правом.

Рисунок 10 – Авторизация

С формы авторизации можно перейти на главную страницу. Для авторизации требуется указать следующие данные: e-mail (уникален в системе), пароль и тип пользователя (студент, преподаватель или администратор).

3.3. Основной функционал системы

3.3.1. Главная страница

После авторизации студент или преподаватель Томского политехнического университета попадает на основную страницу веб-приложения.

Если пользователь не авторизован, доступна только количественная статистика по текущим работам типа «Творческий проект» для 1 и 2 курсов и «Научно-исследовательский проект» для 2, 3 и 4 курсов.

Если пользователь авторизован, дополнительно доступен список текущих работ, отсортированный по дате создания проекта.

Главная страница, как и все остальные, кроме страницы авторизации, имеет шапку (хедер), откуда можно перейти на любую другую страницу, кроме страницы авторизации, если пользователь авторизован, и на любую другую страницу, кроме страницы личного кабинета, если пользователь не авторизован.

Вид главной страницы для авторизованного пользователя представлен на рисунке 11.

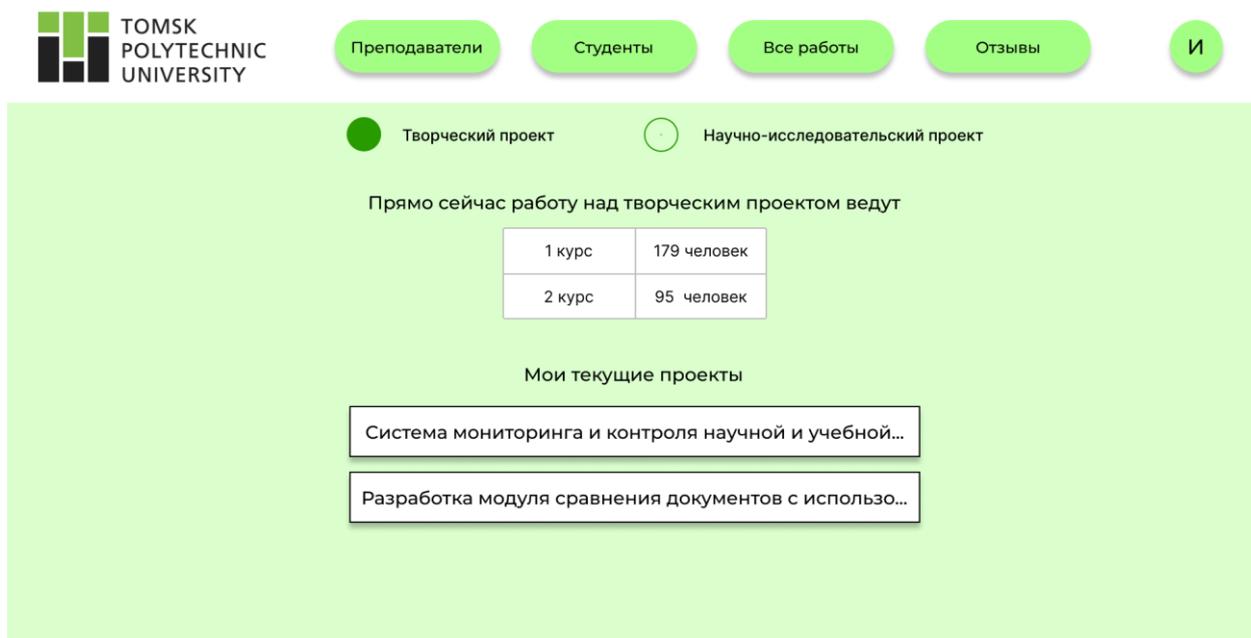


Рисунок 11 – Главная страница

3.3.2. Страница «Преподаватели»

Данная страница предоставляет возможность просмотреть список преподавателей и содержит строку поиска по ФИО преподавателя с функцией автоматической фильтрации.

При выборе конкретного преподавателя на странице отображается следующая информация о преподавателе:

- ФИО;
- отделение;
- должность в университете;
- ученая степень;
- e-mail.

Если роль пользователя – `ROLE_STUDENT`, то дополнительно на странице «Преподаватели» отображается кнопка «Сгенерировать текст

письма», предоставляющая по нажатию автоматически сгенерированный текст электронного письма текущему преподавателю следующего содержания:

- обращение к преподавателю по имени и отчеству;
- перечисление научных специализаций преподавателя;
- просьба взять тему для проекта;
- ФИО, номер группы, номер курса авторизованного студента.

Вид страницы «Преподаватели» для авторизованного студента представлен на рисунке 12.

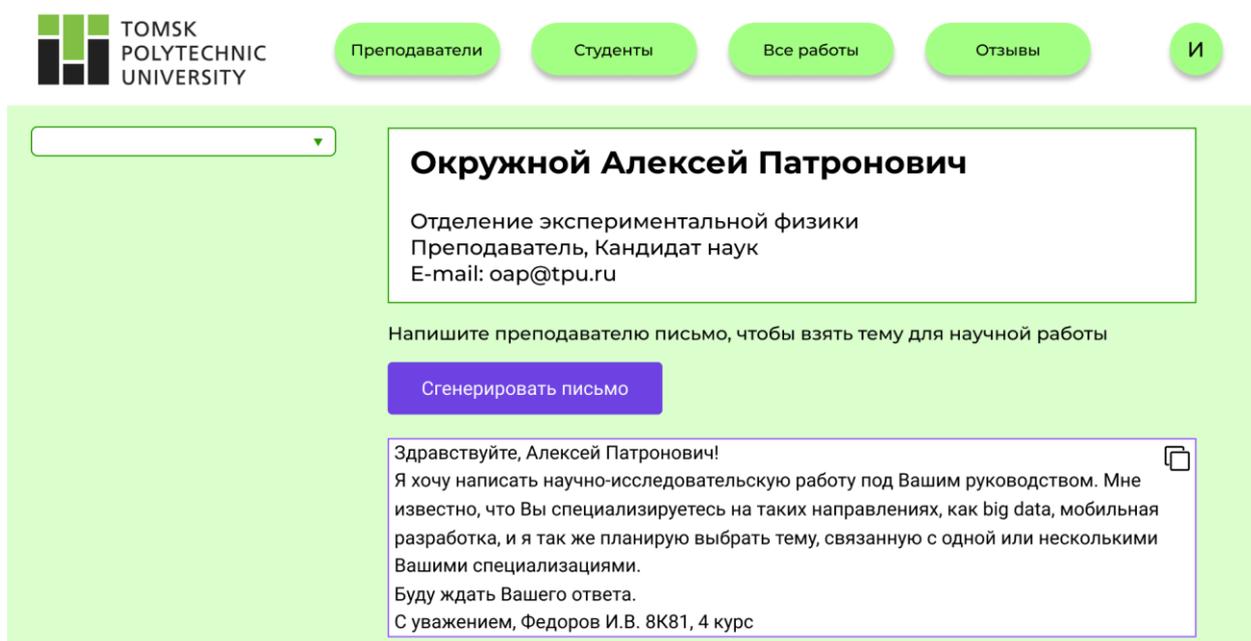


Рисунок 12 – Страница «Преподаватели»

3.3.3. Страница «Студенты»

Данная страница предоставляет возможность просмотреть список студентов и содержит строку поиска по ФИО студента с функцией автоматической фильтрации.

При выборе конкретного студента на странице отображается следующая информация о студенте:

- ФИО;
- номер группы;
- специальность;
- отделение;
- e-mail.

Если студент прикреплен к определенному научному руководителю, то его ФИО будут видны на данной странице

Вид страницы «Студенты» для пользователя с любой ролью представлен на рисунке 13.

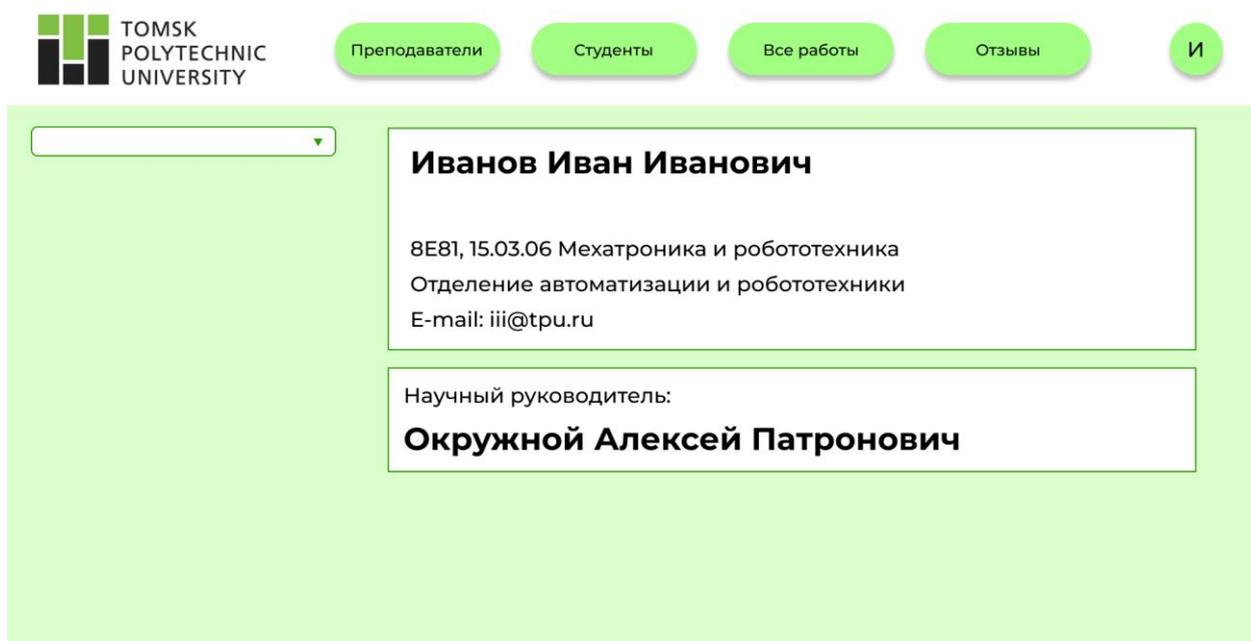


Рисунок 13 – Страница «Студенты»

3.3.4. Страница «Все работы»

Страница «Все работы» в основном предназначена для просмотра и фильтрации списка проектов.

Все пользователи могут просматривать список проектов на данной странице. Информация о проекте включает:

- тему;

- статус (если работа над проектом завершена, отображается дата завершения, иначе – «В разработке»);
- ФИО студентов;
- ФИО научного руководителя.

Дата завершения отображается для определенного проекта, когда атрибут «completion_date» в соответствующей таблице в базе данных принимает значение «null».

Также есть возможность отфильтровать список всех проектов по ФИО научного руководителя (преподавателя), ФИО студента, наименованию темы проекта (части названия) и срокам реализации проекта.

Пользователям, которые являются авторизованными преподавателями либо администраторами, на данной странице дополнительно видна кнопка «Создать проект». По ее нажатию на странице «Все работы» становится видна форма создания нового проекта, где от пользователя требуется следующая информация:

- тема работы;
- тип работы (творческий проект, научно-исследовательский проект);
- ФИО студентов, которые будут ее выполнять (выбираются из списка с автоматической фильтрацией по ФИО).

После этого созданный проект вносится в базу данных с текущей датой начала работы над проектом, и его становится видно в общем списке.

Вид страницы «Все работы» для пользователя с любой ролью представлен на рисунке 14.

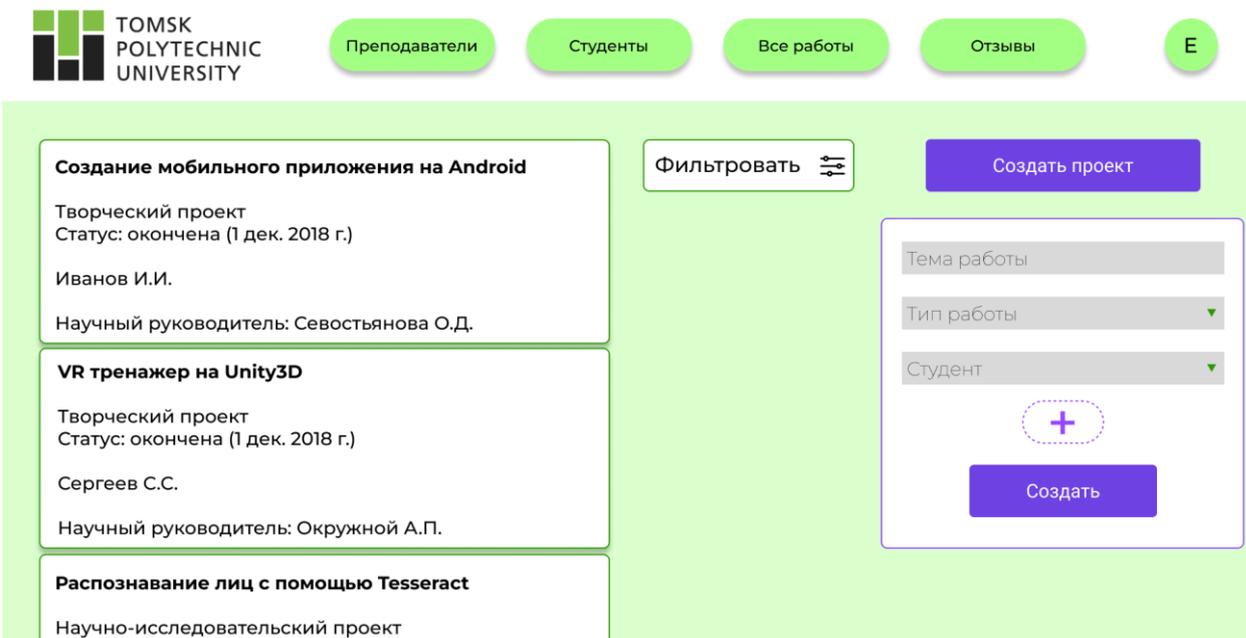


Рисунок 14 – Страница «Все работы»

3.3.5. Страница профиля

Доступ к своему личному кабинету в рамках разрабатываемой информационной системы, безусловно, есть лишь у авторизованного пользователя.

Главным образом, страница профиля содержит данные о пользователе. В нее входит информация, отображаемая на страницах «Преподаватели» и «Студенты», для авторизованного преподавателя и авторизованного студента соответственно.

Ниже основной информации о пользователе находится списки проектов, закрепленных за конкретным пользователем. Это списки текущих и завершенных проектов, где по каждому проекту доступна та же информация, что и на странице «Все работы».

Если пользователь – авторизованный студент, ему видны оценки по каждому завершенному проекту.

Если пользователь – авторизованный преподаватель, на любой текущий проект из списка можно нажать, чтобы его завершить. По нажатию пользователю предлагается заполнить форму завершения проекта, где предлагается заполнить следующие поля:

- оценка каждому студенту;
- дата завершения, если нужна конкретная дата;
- комментарий (необязательно).

Таблицы в базе данных обновляются по измененному проекту.

В самом низу страницы профиля находится кнопка «Выйти».

Вид страницы профиля для авторизованного преподавателя представлен на рисунке 15.

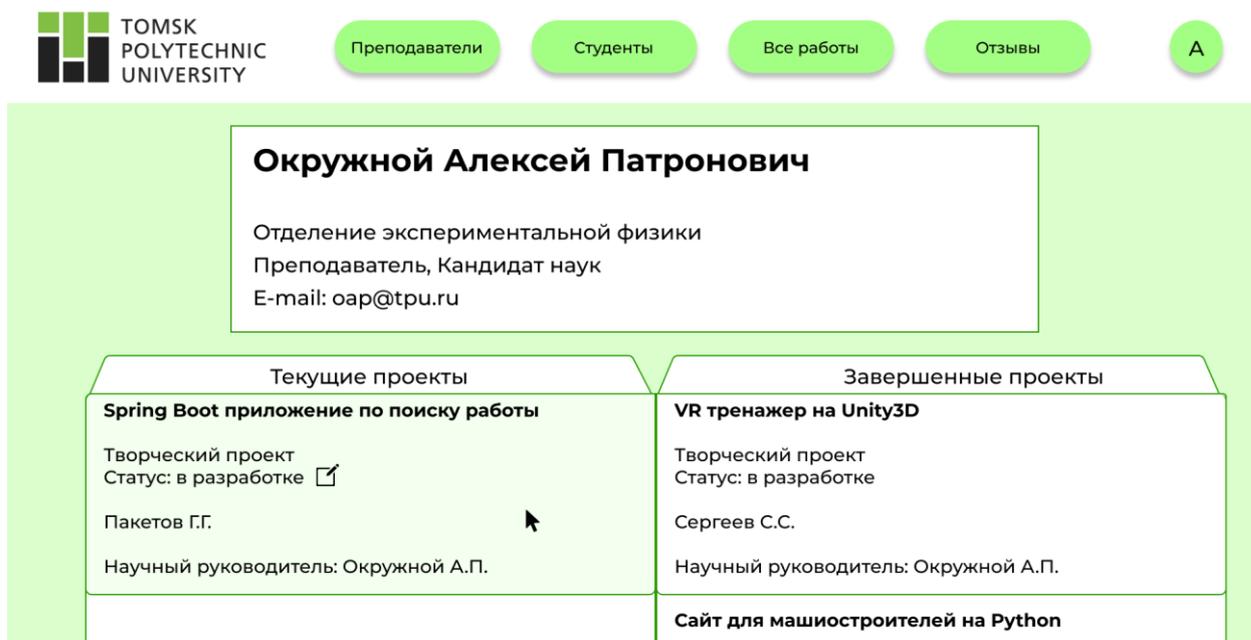


Рисунок 15 – Страница профиля

3.3.6. Страница «Отзывы»

На данной странице находится список отзывов, отсортированных по дате создания отзыва и содержащих следующую информацию:

- ФИО и e-mail автора отзыва;

- ФИО пользователя, кому отзыв адресован;
- текст отзыва.

Если пользователь не является неавторизованным, то ему видна кнопка «Создать отзыв», предлагающая заполнить ФИО пользователя, кому отзыв адресован и текст отзыва, прежде чем окончательно нажать кнопку «Создать». После создания отзыва, он добавляется в базу данных и становится виден в общем списке на данной странице.

Также присутствует строка поиска отзывов по ФИО студентов и преподавателей. Вид страницы «Отзывы» для авторизованного пользователя представлен на рисунке 16.

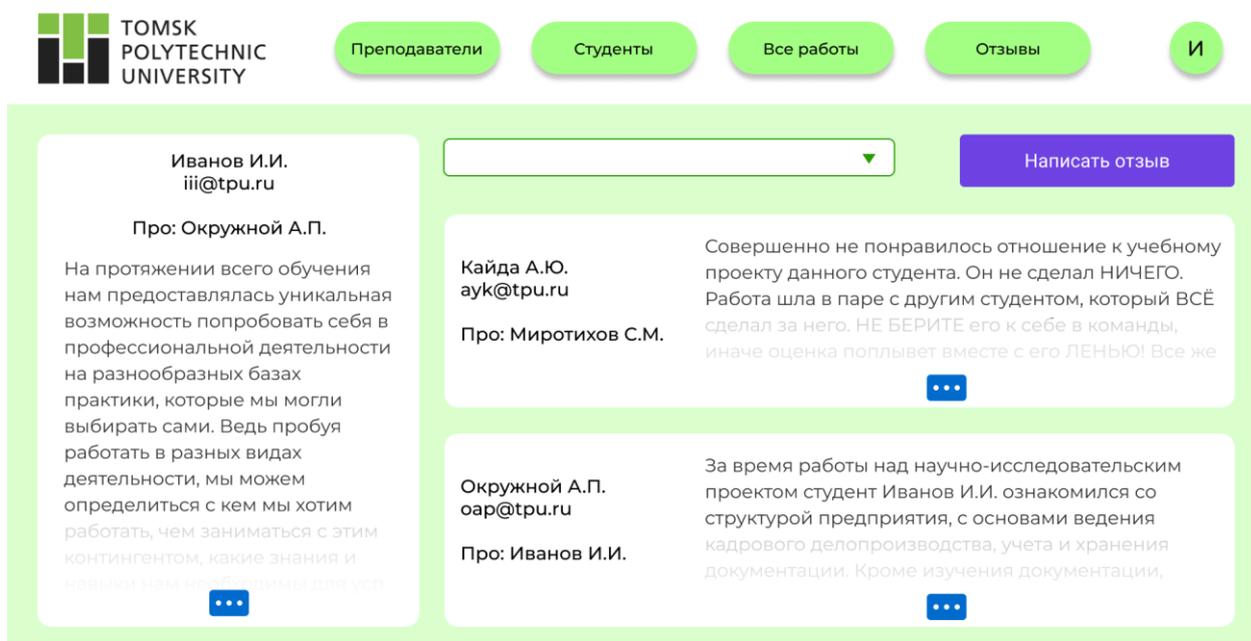


Рисунок 16 – Страница «Отзывы»

3.4. Вывод по разделу

На данном этапе была описана функциональность каждой из семи страниц разрабатываемой системы.

4. Разработка серверной части разрабатываемой системы

4.1. Описание технологии разработки веб-приложения

Система является веб-приложением – из соображений доступа к ней с любого устройства и в любое время.

Серверная часть приложения разрабатывалась с использованием универсального фреймворка с открытым исходным кодом для Java-платформы – Spring Framework, а именно – Spring Boot [4].

Spring Boot – это утилита, которая обладает большим функционалом. Наиболее важными особенностями Spring Boot являются:

- **Автоматическая конфигурация.** Без использования Spring Boot конфигурация таких частей веб-приложения, как распознаватель представлений, драйвер для работы с базами данных, распознаватель шаблонов и другие, требует написания немалого количества строк Java или XML кода на самом примитивном уровне. В Spring Boot появляется возможность определять встроенные и кастомизированные настройки – properties – и внедрение их в код с использованием синтаксиса заполнителей [5] и метаданных (далее – аннотаций) [6]. Помимо считывания информации, Spring Boot автоматически регистрирует бины – Java-объекты, чей жизненный цикл управляется с помощью контейнера Spring – исходя из добавленных зависимостей. Автоматическая конфигурация может в любой момент быть переопределена в соответствии с пользовательскими настройками.
- **Управление зависимостями.** Spring Boot неявно упаковывает необходимые сторонние зависимости для веб-приложения на основе Spring и предоставляет их посредством так называемых starter-пакетов.

- Встроенная поддержка сервера приложений – контейнера сервлетов. Сервлеты используются для расширения веб-серверов, взаимодействуя с клиентами по принципу вопрос-ответ. Веб-сервер встроен в каждое веб-приложение, написанное с использованием Spring Boot. Также доступны определенные starter-пакеты для различных HTTP-серверов. Это полезно в разработке микросервисных Spring Boot приложений: установка и развертывание возможна для любого сервиса в полностью автономном артефакте, который также предоставляет API.

4.2. Разработка веб-приложения на Spring Boot

4.2.1. Подключение к базе данных MySQL

База данных к разрабатываемой информационной системе создавалась с использованием СУБД MySQL. Подключение к базе данных выполнялось в среде разработки IntelliJ IDEA путем внедрения зависимости для MySQL Connector и считывания кастомизированных пользовательских настроек и конфигурации JPA [7] (Приложение Б).

JPA (Java Persistence API) – это спецификация, описывающая систему управления сохранением Java-объектов в таблицы реляционных баз данных в удобном виде. Одной из самых популярных реализаций последней версии JPA является Hibernate [8].

Hibernate относят к разновидности ORM (Object-Relational Mapping), выступающей в качестве связующего звена между веб-приложением и системой управления базами данных. Таким образом, создается виртуальная база данных, в которой находится скрытая логика по управлению запросами.

4.2.2. Работа Hibernate при обмене данных

ORM на высоком уровне совершает следующие действия по запросу типа SELECT (на примере Hibernate и MySQL):

1. Определяет логику конкретного Java-метода, например, `findAllStudents()`.
2. Посылает SQL запрос «`SELECT * FROM Student ...`» в MySQL.
3. Получает результат запроса и предоставляет его серверу веб-приложения в заранее известном, сконфигурированном разработчиком виде.

Для данного примера в разрабатываемой системе, мы получили список студентов, который может не подвергаться никаким мутациям или, наоборот, может быть изменен преподавателем, который выставляет оценки.

Для реализации функционала JPA, был создан интерфейс, реализующий интерфейс `JpaRepository` (Приложение В). Чтобы определить SQL для выполнения метода, мы можем пометить его аннотацией `@Query` – его атрибут `value` содержит JPQL или SQL для выполнения. Определение запроса по умолчанию использует JPQL. Аннотацию можно опустить при соблюдении определенного синтаксиса [9] при написании сигнатуры аннотированного метода.

Созданный интерфейс может реализовать класс, аннотированный как «`@Repository`» или «`@Service`». `Repository`-аннотация на классе означает, что в нем сосредоточена `crud`-логика (`Create`, `Read`, `Update`, `Delete`). `Service`-аннотацией помечаются классы с сосредоточением остальной необходимой бизнес-логики.

4. Логично было бы предположить, что `Hibernate` должен посылать модифицирующий SQL запрос, например, «`UPDATE student SET ...`» в соответствии с вызванным пользователем методом, изменяющим данные: `saveStudents(Collection<Student> c)`.

Такой метод репозитория отличается от запроса на выборку тем, что он обязательно должен быть аннотирован как «`@Modifying`».

Однако такие изменения данных могут конфликтовать с изменениями кого-то, кто параллельно работал с этой базой данных и с этими же самыми студентами.

По этой причине современные ORM отправляют СУБД не модифицирующий запрос, а тот же «SELECT * FROM student ...», ожидая такой же набор объектов, как в 3 пункте. Внутри Hibernate находится жесткая логика, которая фактически позволяет определить, есть конфликт или нет. Если конфликтов не обнаружено, проводится ожидаемая клиентом модификация данных. При наличии неразрешенного конфликта ORM не может устранить его на своем уровне и перекладывает ответственность на разработчика и прикладную логику. Это так же правильно, как не доверять системе управления версиями, например, Git, самой принимать решения в неоднозначных ситуациях, например, при merge. При возникновении конфликта выбрасывается исключение, и мутация данных в реальной базе данных не происходит.

Важно, что между двумя описанными запросами выборки никакой транзакции организовано не было. Иначе за это время никакой другой пользователь с этим же списком студентов работать бы не смог.

Описанное поведение Hibernate по своей архитектуре напоминает архитектуру CAS (compare and set/swap) [10] на очень высоком уровне, где текущее значение – это второй запрос на выборку – сравнивается с ожидаемым значением, то есть первым запросом, до проведения модификаций с целью валидации.

Схема работы Hibernate в разрабатываемой информационной системе изображена на рисунке 17.

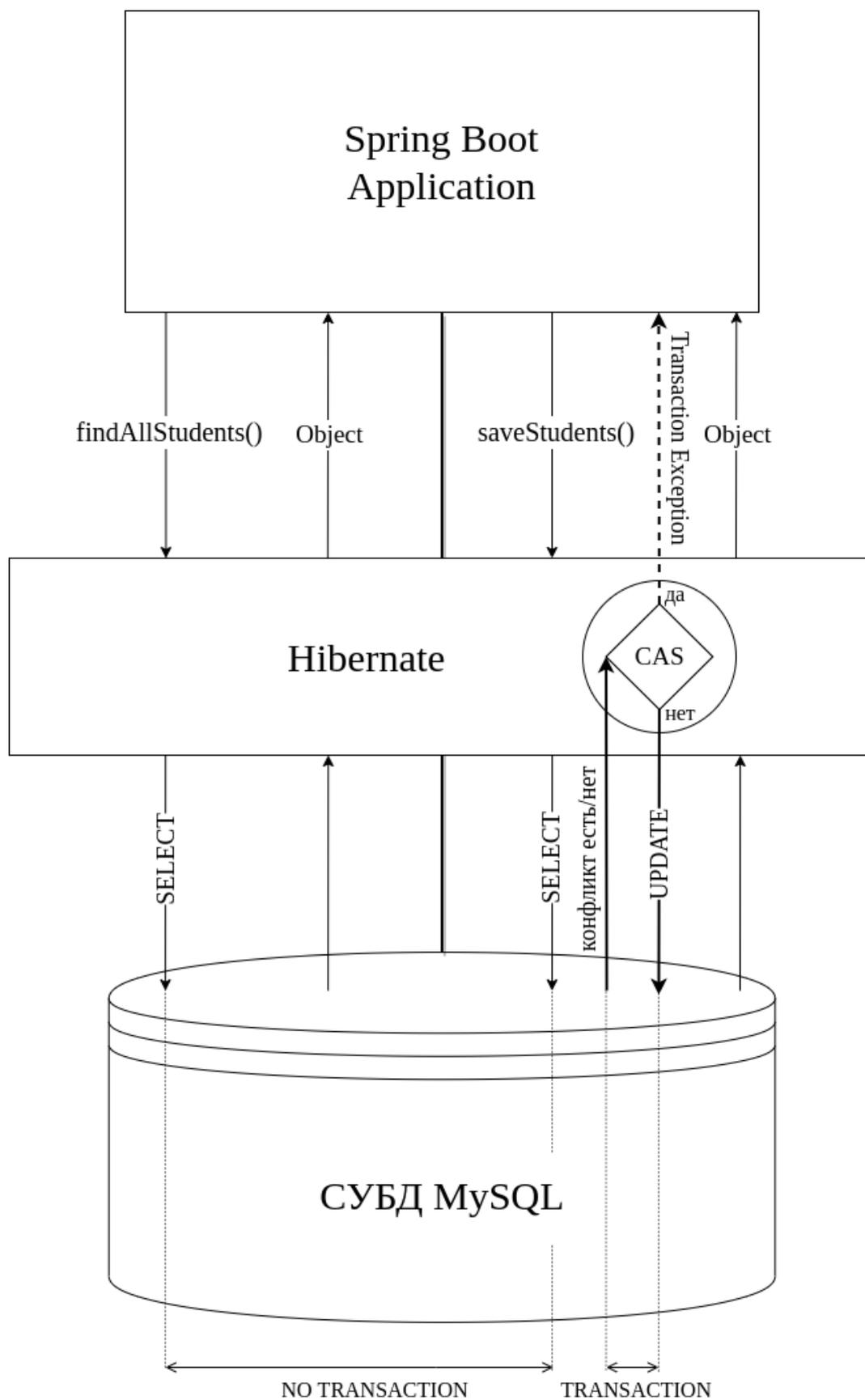


Рисунок 17 – Работа Hibernate при выборке данных и их модификации

4.2.3. Создание условий для работы с ORM с использованием JPA

Легковесным хранимым объектом бизнес-логики разрабатываемого веб-приложения с базой данных является Entity (далее – класс сущности). Это специальный Java-класс, аннотируемый как «@Entity» и содержащий переменные в соответствии с атрибутами определенной таблицы в базе данных.

Нibernate обращается к классу виртуальной базе данных при формировании JPQL запросов. Важно верно нормализовать модель данных заранее, чтобы Java Persistence API работал исправно.

При написании классов сущностей учитывались следующие детали:

- состав первичного ключа: составной или нет;
- логический смысл внешнего ключа: для определения типа ссылочного объекта;
- тип связи: один к одному, один ко многим, многие к одному, многие ко многим через другую таблицу (при этом не обязательно указывать, идентифицирующая ли связь);
- назначение соответствующей таблицы в базе данных: отдельная сущность, связующая таблица.

При наличии составного первичного ключа у какой-либо таблицы этот ключ размещался в виде отдельного неавтономного встроенного класса с аннотацией «@Embeddable», а новое поле идентификатора класса сущности было аннотировано как «@EmbeddedId».

Независимо от типа связи, при наличии внешнего ключа могли создаваться дополнительные поля в классе сущности, не входящие в таблицу в реальной базе данных. Такие поля аннотировались как «@Transient», чтобы они игнорировались при передаче данных СУБД.

Поля класса сущности, по атрибутам которых связаны таблицы, были также дополнительно настроены через аннотации «@JoinColumn» и «@JoinTable», которые поддерживают различное количество аргументов в своей сигнатуре, что позволяет регулировать тип связи в соответствии с моделью данных, иногда в связке с другими аннотациями, явно маркирующие тип связи.

В Приложении Г приведен код для классов сущностей Teacher, Science_specialization и Teacher_specialization и продублированы соответствующие таблицы из логической модели данных.

4.2.4. Подключение Spring Security для управления доступом к данным

Главная цель использования Spring Security – ограничение доступа к различным страницам сайта для пользователей с разными ролями.

В веб-приложении реализована авторизация и ограничение доступа к страницам сайта в зависимости от роли пользователя: неавторизованный пользователь (ROLE_USER), авторизованный студент (ROLE_STUDENT), авторизованный преподаватель (ROLE_TEACHER) и администратор (ROLE_ADMIN).

В Приложении Д приведен код конфигурации безопасности, прописанной в классе «WebSecurityConfig». Метод configure() включает в себя настройку доступа к ресурсам сайта, то есть конфигурацию путей к страницам или файлам, для которых необходимо установить ограничение, а также список ролей по каждому ограничению.

4.2.5. Использование MVC при разделении компонентов программы

Архитектура MVC позволяет разделить код веб-приложения на 3 части: Модель (Model), Вид или Представление (View) и Контроллер (Controller).

Контроллер взаимодействует с Представлением, или наоборот, с помощью Модели. Для обеспечения этого взаимодействия был использован Thymeleaf.

Thymeleaf – это шаблонизатор Java HTML. Он предназначен для обслуживания HTML5 на уровне представления веб-приложений на основе MVC. В данном проекте Thymeleaf используется для удобства шаблонизации и разметки представлений.

В Приложении Е продемонстрирован код контроллера для страницы «Все работы». Аннотациями «@GetMapping» и «@PostMapping» были помечены методы, которые обрабатывают HTTP-GET и HTTP-POST запросы соответственно по данному выражению URI.

С помощью модели можно обмениваться любыми типами данных в удобном виде как для клиента, так и для разработчика. В Приложении Е двумя способами показана передача данных Модели из класса Контроллера: явно с помощью метода `addAttribute()` интерфейса Модели и с помощью аннотации «@ModelAttribute».

Передача данных Представления Контроллеру была реализована как через тег `<form>` и синтаксис Thymeleaf, так и с помощью JavaScript. В Приложении Ж представлена часть Представления для страницы «Все работы». Фильтрация по теме работы была реализована явно с помощью Модели и Thymeleaf. Фильтрация по ФИО студента была реализована с использованием JavaScript.

При загрузке страницы Все работы можно просматривать генерацию JPQL запросов Hibernate к MySQL в консоли развертывания Spring Boot веб-приложения, обозначив в файле `application.properties` следующий флаг: `spring.jpa.show-sql=true`. В Приложении И представлены Hibernate запросы при загрузке страницы «Работы».

4.3. Вывод по разделу

В рамках данного раздела были описаны особенности разработки серверной части информационной системы: тип системы (веб-приложение), фреймворк (Spring Boot), особенности работы с ORM (JPA и Hibernate) и использование шаблона MVC при написании Контроллеров и Представлений.

5. Финансовый менеджмент, ресурсоэффективность и ресурсосбережение

Проектная группа по разработке и реализации проекта состоит из двух лиц: научный руководитель и разработчик.

Данная выпускная квалификационная работа заключается в проектировании и разработке веб-приложения отслеживания работ и тем исследовательской деятельности каждого студента вуза и руководителей. В рамках данной работы реализуется веб-приложение и его основные модули-сервисы.

Целью раздела «Финансовый менеджмент, ресурсоэффективность и ресурсосбережение» является выбор наиболее конкурентоспособных методологий разработки, оценка эффективности, определение рисков и стратегий их устранения, формирование состава работ и бюджета проекта.

Для достижения поставленной цели были сформулированы следующие задачи:

1. Проанализировать альтернативные варианты реализации проекта.
2. Оценить коммерческий потенциал и перспективность разработки проекта.
3. Провести оценку научно-технического уровня исследования и оценку рисков.
4. Составить план работ по реализации проекта.
5. Рассчитать бюджет проекта.

5.1. Оценка коммерческого потенциала и перспективности научных исследований

5.1.1. Потенциальные потребители результатов исследования

В данной выпускной квалификационной работе рассматривается разработка системы мониторинга и контроля научной и учебной исследовательской деятельности студента. Отслеживание исследовательской активности необходимо как управляющим отделам организации, так и преподавателям и студентам.

На основе регулярно собираемых разработанной информационной системой данных о студентах и преподавателях научно-образовательной организации и их проектной деятельности ответственные за исследовательскую деятельность лица смогут грамотно оценивать текущее положение дел и принимать управленческие решения в области научной и учебной исследовательской деятельности для поддержания уровня и развития организации.

Преподаватели смогут наблюдать за успехами студентов и, при необходимости, находить тех, кто удовлетворяет определенным запросам для дальнейшей совместной работы в предстоящем семестре и в будущем. Студенты смогут выбирать научного руководителя для учебных и научных исследовательских работ, исходя из информации об их научных специализациях.

В таблице 5.1 представлена карта сегментирования рынка услуг по разработке веб-приложения, предоставляющего данные о научной и учебной исследовательской деятельности в научно-образовательной организации.

Таблица 5.1 – Карта сегментирования рынка услуг по разработке веб-приложения, предоставляющего данные о научной и учебной исследовательской деятельности в научно-образовательной организации

	Тип платформы
--	---------------

		Мобильное приложение	Веб-приложение	Телеграм-бот
Данные	О проектной деятельности	СР	ОСПР	С
	О студентах и преподавателях	СП	ОСПР	СПР

С – студенты, П – преподаватели, О – ответственные за исследовательскую деятельность, Р – рядовые пользователи

На карте сегментирования видно, что мобильными приложениями и Телеграм-ботами с целью получить определенные данные пользуются не все, а веб-приложения подходят любым типам пользователей. Именно на этот сегмент стоит ориентироваться.

5.1.2. Анализ конкурентных технических решений

В выборе наиболее эффективного подхода к реализации проекта анализ конкурентных технических решений очень важен. Необходимо учесть все сильные и слабые стороны продукта для того, чтобы всегда быть востребованным на рынке, развивая конкурентные преимущества собственного продукта.

Для оценки конкурентоспособности разрабатываемого веб-приложения была составлена карта сравнения конкурентных технических решений, представленная в таблице 5.2. Индексом «в» обозначена собственная разработка, индексом «п» – техническое решение сайта ТПУ совместно с Порталом ТПУ, индексом «г» – техническое решение сайта ТГУ.

Таблица 5.2 – Оценочная карта для сравнения конкурентных разработок

Критерии оценки	Вес критерия	Баллы			Конкурентоспособность		
		Б _в	Б _п	Б _г	К _в	К _п	К _г
1	2	3	4	5	6	7	8

Технические критерии оценки эффективности							
1. Удобство интерфейса	0,1	3	5	5	0,3	0,5	0,5
2. Простота поддержки	0,1	5	3	3	0,5	0,3	0,3
3. Отказоустойчивость	0,1	4	4	4	0,4	0,4	0,4
4. Информативность в рамках проектной деятельности	0,2	5	3	2	1	0,6	0,4
5. Наглядность	0,15	4	2	2	0,6	0,3	0,3
6. Масштабируемость	0,15	4	2	2	0,6	0,3	0,3
Экономические критерии оценки эффективности							
7. Цена обслуживания	0,1	4	5	5	0,4	0,5	0,5
8. Интерактивность	0,1	4	5	5	0,4	0,5	0,5
Итого	1	3 3	2 9	28	4,2	3,4	3,2

Анализ конкурентных технических решений определяется по формуле:

$$K = \sum V_i \times B_i \quad (5.1)$$

где K – конкурентоспособность вида;

V_i – вес критерия (в долях единицы);

B_i – балл i -го показателя.

По результатам, которые предоставляет оценочная карта можно сделать вывод, что наиболее конкурентоспособным решением будет разработка собственного веб-приложения. Основными преимущественными

факторами являются интерактивность, наглядность и масштабируемость, совмещенных в едином проекте.

5.1.3. Технология QuaD

Для того, чтобы оценить перспективность разработки в соответствие с методологией QuaD была составлена оценочная карта, представленная в таблице 5.3. При составлении оценочной карты для каждого критерия был определён его вес таким образом, чтобы сумма всех весов равнялась единице.

Таблица 5.3 – Оценочная карта перспективности разработки

Критерии оценки	Вес критерия	Балл	Мак балл	Относительное значение	Средне-взвешенное значение
Показатели оценки качества разработки					
Потребность в ресурсах памяти	0,01	60	100	0,6	0,06
Функциональная мощность	0,01	80	100	0,8	0,08
Простота эксплуатации	0,012	90	100	0,9	0,0108
Качество графического интерфейса	0,01	80	100	0,8	0,008
Скорость работы	0,065	80	100	0,8	0,052
Энергоэффективность	0,05	80	100	0,8	0,004
Помехоустойчивость	0,05	60	100	0,6	0,03
Надёжность	0,05	60	100	0,6	0,03
Унифицированность	0,03	100	100	1	0,03
Уровень материалоемкости разработки	0,05	80	100	0,8	0,04
Уровень шума	0,01	100	100	1	0,01
Безопасность	0,05	60	100	0,6	0,03
Потребность в ресурсах памяти	0,03	80	100	0,08	0,024
Функциональная мощность	0,06	60	100	0,6	0,036
Простота эксплуатации	0,06	100	100	1	0,06
Качество интеллектуального интерфейса	0,06	80	100	0,8	0,048
Ремонтопригодность	0,01	80	100	0,8	0,008
Показатели оценки экономического потенциала разработки					

Конкурентоспособность продукта	0,08	100	100	1	0,08
Уровень проникновения на рынок	0,08	20	100	0,2	0,0016
Перспективность рынка	0,06	100	100	1	0,06
Цена	0,03	80	100	0,8	0,024
Послепродажное обслуживание	0,06	60	100	0,6	0,036
Финансовая эффективность разработки	0,04	40	100	0,4	0,016
Срок выхода на рынок	0,05	60	100	0,6	0,03
Наличие сертификации разработки	0,07	20	100	0,2	0,014
Итого	1				0,82

Анализ, выполненный по технологии QuaD, показал, что разработка данного веб-приложения является перспективной, так как итоговое показательное значение 82.

5.1.4. SWOT-анализ

Для выявления всевозможных факторов, которые могут повлиять на успешность продукта в условиях рынка, необходимо следующим шагом провести SWOT-анализ. Результатом этого анализа будет стратегия дальнейшего развития и продвижения продукта на рынке. Также, посредством SWOT-анализа определяются сильные и слабые стороны проекта, а также возможности и угрозы.

Сильные стороны — это те факторы, которые характеризуют конкурентоспособность проекта.

Слабые стороны — это ограничения в возможностях или ресурсах, которые есть у проекта.

Возможности — это благоприятные характеристики внешней среды, которые можно использовать для составления успешной стратегии реализации проекта.

Угрозы — это негативные факторы, которые в настоящем или будущем могут стать барьером для успешности проекта.

Первый этап SWOT-анализа состоит из выявления сильных и слабых сторон, возможностей и угроз. Результаты первого этапа представлены в таблице 5.4.

Таблица 5.4 – Матрица SWOT анализа

Сильные стороны	Возможности во внешней среде
<p>С1. Отсутствие идентичных сервисов;</p> <p>С2. Гибкая и мощная архитектура системы;</p> <p>С3. Стабильность и скорость работы системы.</p>	<p>В1. Предоставление полной информации о проектной деятельности в организации и формирование статистики;</p> <p>В2. Высокий спрос на разработку новых сервисов, которым необходима платформа;</p> <p>В3. Использование современных инструментов разработки.</p>
Слабые стороны	Угрозы внешней среды
<p>Сл1. Отсутствие коммерческого опыта реализации веб-приложения;</p> <p>Сл2. Большие временные затраты;</p> <p>Сл3. Сложности с модерированием системы.</p>	<p>У1. Несоответствие требованиям пользователей;</p> <p>У2. Нехватка специалистов для дальнейшей поддержки проекта;</p> <p>У3. Нехватка финансирования.</p>

Второй этап состоит в построении интерактивной матрицы проекта. Она позволяет оценить взаимосвязь факторов, чтобы сформировать или скорректировать стратегию. Каждый фактор помечается либо знаком «+»

(означает сильное соответствие сильных сторон возможностям), либо знаком «-» (что означает слабое соответствие); «0» – если есть сомнения в том, что поставить «+» или «-». Интерактивная матрица сильных и слабых сторон и возможностей проекта представлена в табл. 5.5. Интерактивная матрица сильных и слабых сторон и угроз проекта представлена в табл. 5.6.

Таблица 5.5 – Интерактивная матрица сильных и слабых сторон и возможностей

Возможности проекта	Сильные стороны			Слабые стороны		
		С1	С2	С3	Сл1	Сл2
B1	+	+	+	-	+	+
B2	+	-	+	+	-	-
B3	+	-	-	-	-	-

Таблица 5.6 – Интерактивная матрица сильных сторон и слабых сторон и угроз

Угрозы проекта	Сильные стороны			Слабые стороны		
		С1	С2	С3	Сл1	Сл2
У1	-	-	-	-	+	+
У2	+	-	+	+	+	+
У3	+	+	+	-	-	-

Корреляцию возможностей и угроз с сильными и слабыми сторонами можно записать в следующем виде:

- B1B2B3C1; B1C2; B1B2C3;
- B1Сл2Сл3; B2Сл1;
- У2У3С1С3; У3С2;
- У1У2Сл2Сл3; У2Сл1

Такой метод записи помогает выявить группы факторов, которые имеют единую природу.

На третьем этапе составляется итоговая матрица SWOT-анализа. В ней из результатов прошлого этапа формируются основные стратегии, которые позволят снизить риски. Итоговая матрица представлена в таблице 5.7.

Таблица 5.7 - Итоговая матрица SWOT-анализа

	<p>Сильные стороны:</p> <p>С1. Отсутствие идентичных сервисов;</p> <p>С2. Гибкая и мощная архитектура системы;</p> <p>С3. Стабильность и скорость работы системы.</p>	<p>Слабые стороны:</p> <p>Сл1. Отсутствие коммерческого опыта реализации веб-приложения;</p> <p>Сл2. Большие временные затраты;</p> <p>Сл3. Сложности с модерированием системы.</p>
<p>Возможности:</p> <p>В1. Предоставление полной информации о проектной деятельности в организации и формирование статистики;</p> <p>В2. Высокий спрос на разработку новых сервисов, которым необходима платформа;</p> <p>В3. Использование современных инструментов разработки.</p>	<p>Сильные стороны и возможности в совокупности говорят о том, что разработка веб-приложения позволит разработать систему, которая будет пользоваться спросом как со стороны студентов, так и сотрудников научно-образовательной организации.</p>	<p>Оценивая возможности и слабые стороны проекта, будет уместным сделать вывод, что система может быть не разработана вовремя.</p>
<p>Угрозы внешней среды:</p> <p>У1. Несоответствие требованиям пользователей;</p> <p>У2. Нехватка специалистов для дальнейшей поддержки проекта;</p> <p>У3. Нехватка финансирования.</p>	<p>Сильные стороны с угрозами проекта говорят о том, что несмотря на отсутствие похожего сервиса, проект может иметь уязвимости, ввиду отсутствия опыта коммерческой разработки.</p>	<p>Сложности с модерированием система вместе с нехваткой специалистов для дальнейшей поддержки проекта сигнализируют о том, что сервис может использоваться нежелательными или неправомерными организациями.</p>

5.2. Определение возможных альтернатив проведения научных исследований

Определим возможные альтернативы проведения с помощью морфологического подхода. Он основан на систематическом исследовании всех теоретически возможных вариантов, вытекающих из закономерностей строения (морфологии) объекта исследования. Синтез охватывает как известные, так и новые, необычные варианты, которые при простом переборе

могли быть упущены. Путем комбинирования вариантов получают большое количество различных решений, ряд которых представляет практический интерес.

В качестве исследуемой предметной области возьмем разработку веб-приложения. В качестве характеристик выделим следующие:

- Тип платформы.
- Инструмент разработки серверной части.
- Фреймворк для разработки серверной части.
- Основной шаблон проектирования серверной части.
- База данных.
- Формат запросов.

В таблице 5.8 приведена морфологическая матрица по разработке веб-приложения.

Таблица 5.8 - Морфологическая матрица по разработке веб-приложения

	1	2	3
А. Тип платформы	Веб-приложение	Мобильное приложение	Telegram-бот
Б. Инструмент разработки серверной части	Java	C#	Python
В. Фреймворк для разработки серверной части	Spring Framework	.NET	Django
Г. Основной шаблон проектирования серверной части	MVC	Шардинг	-
Д. База данных	PostgreSQL	MySQL	MongoDB

Е. Формат запросов	JSON RPC	REST API	GraphQL
--------------------	----------	----------	---------

Основным вариантом решения будет А1Б1В1Г1Д2Е2.

Данный вариант позволит сэкономить время на разработку, так как имеется опыт работы на Java с использованием Spring Boot

Альтернативными вариантами решения будут:

- А1Б1В1Г1Д3Е2 – позволит разработать веб-приложение с использованием не реляционной системы управления базами данных при взаимодействии с Java;
- А1Б2В2Г2Д2Е2 – позволит применить другую популярную технологию разработки (С# и .NET).

5.3. Планирование работ по научно-техническому исследованию

5.3.1. Структура работ в рамках научного исследования

Предполагаемые работы, которые необходимо осуществить, спланированы следующим образом:

- определение структуры работ в рамках научного исследования;
- определение участников каждой работы;
- установление продолжительности работ;
- построение графика проведения научных исследований.

Для проведения научных исследований формируется рабочая группа, в состав которой могут входить научные сотрудники и преподаватели, разработчики, инженеры, техники и лаборанты, численность групп может варьироваться. По каждому виду запланированных работ устанавливается соответствующая должность исполнителей.

В данном разделе необходимо составить перечень этапов и работ в рамках проведения научного исследования, провести распределение

исполнителей по видам работ. Перечень этапов и работ, распределение исполнителей по данным видам работ представлен в таблице 5.9.

Таблица 5.9 – Перечень этапов, работ и распределение исполнителей

Основные этапы	№ раб	Содержание работ	Должность исполнителя
Разработка технического задания	1	Составление и утверждение технического задания	Руководитель
Выбор направления исследований	2	Выбор направления исследований	Разработчик
	3	Подбор и изучение материалов по теме	Разработчик
	4	Календарное планирование работ	Руководитель Разработчик
Теоретические и экспериментальные исследования	5	Исследование методологии построения архитектуры системы	Руководитель Разработчик
	6	Концептуализация системы	Разработчик
Обобщение и оценка результатов	7	Оценка эффективности полученных результатов	Разработчик
	8	Определение целесообразности проведения ОКР	Разработчик
Проведение ОКР			
Разработка технической документации и проектирование	9	Проектирование системы и составление диаграмм	Разработчик
	10	Оценка эффективности разработки и применения системы	Разработчик
Изготовление и испытание макета (опытного образца)	11	Разработка системы	Разработчик
	12	Тестирование разработанной системы	Разработчик

	13	Развёртывание системы	Разработчик
Оформление отчета по НИР и комплекта документации по ОКР	14	Оформление эксплуатационно-технической документации	Разработчик
	15	Составление пояснительной записки	Разработчик

5.3.2. Определение трудоемкости выполнения работ

Трудовые затраты в большинстве случаев образуют основную часть стоимости разработки, поэтому важным моментом является определение трудоемкости работ каждого из участников научного исследования.

Трудоемкость выполнения научного исследования оценивается экспертным путем в человеко-днях и носит вероятностный характер, который зависит от множества трудно учитываемых факторов. Для определения ожидаемого (среднего) значения трудоемкости $t_{ожi}$ используется следующая формула:

$$t_{ожi} = \frac{3t_i + 2t_i}{5}, \quad (5.2)$$

где $t_{ожi}$ – ожидаемая трудоемкость выполнения i -ой работы чел.-дн.;

t_i – минимально возможная трудоемкость выполнения заданной i -ой работы, чел.-дн.;

t_i – максимально возможная трудоемкость выполнения заданной i -ой работы, чел.-дн.;

Исходя из ожидаемой трудоемкости работ, определяется продолжительность каждой работы в рабочих днях T_p , учитывающая параллельность выполнения работ по нескольким исполнителями.

$$T_{pi} = \frac{t_{ожi}}{ч_i}, \quad (5.3)$$

где T_{pi} – продолжительность одной работы, раб.дн.;

$t_{ожi}$ – ожидаемая трудоемкость выполнения одной работы, чел.-дн.;

$Ч_i$ – численность исполнителей, выполняющих одновременно одну и ту же работу на данном этапе, чел.

5.3.3. Разработка графика проведения научного исследования

Диаграмма Ганта – горизонтальный ленточный график, на котором работы по выбранной теме изображены протяженными отрезками, представляющие собой временные промежутки, имеющие даты начала и окончания выполнения работ.

Данный вид диаграммы удобен и нагляден для представления проведения научных работ.

Для удобства построение графика, длительность каждого из этапов работ из рабочих дней следует перевести в календарные дни. Для этого необходимо воспользоваться следующей формулой:

$$T_{ki} = T_{pi} \cdot k_{\text{кал}}, \quad (5.4)$$

где T_{ki} – продолжительность выполнения i -й работы в календарных днях;

T_{pi} – продолжительность выполнения i -й работы в рабочих днях;

$k_{\text{кал}}$ – коэффициент календарности.

Коэффициент календарности определяется по следующей формуле:

$$k_{\text{кал}} = \frac{T_{\text{кал}}}{T_{\text{кал}} - (T_{\text{вых}} + T_{\text{пр}})}, \quad (5.5)$$

где $T_{\text{кал}}$ – количество календарных дней в году;

$T_{\text{вых}}$ – количество выходных дней в году;

$T_{\text{пр}}$ – количество праздничных дней в году.

Расчет коэффициента календарности:

$$k_{\text{кал}} = \frac{T_{\text{кал}}}{T_{\text{кал}} - (T_{\text{вых}} + T_{\text{пр}})} = \frac{365}{365 - 118} = 1,48$$

Таблица 5.10 – Временные показатели проведения научного исследования

Название работы	Трудоёмкость работ						Исполнители	Длительность работ в рабочих днях T_{pi}		Длительность работ в календарных днях T_{ki}	
	T_{min} , чел–дни		T_{max} , чел–дни		$T_{ожи}$, чел– дни			Исп.1	Исп.2	Исп.1	Исп.2
	Исп.1	Исп.2	Исп.1	Исп.2	Исп.1	Исп.2					
Составление и утверждение ТЗ	2	2	3	3	2,4	2,4	Научный руководитель	2	2	2	2
Выбор направлений исследований	1	1	2	2	1,4	1,4	Научный руководитель, разработчик	1	1	1	1
Подбор и изучение материалов по теме	2	2	4	4	2,8	2,8	Разработчик	1	1	1	1
Календарное планирование работ	1	1	2	2	1,4	1,4	Научный руководитель, разработчик	1	1	1	1
Исследование методологии построения архитектуры системы	4	3	6	5	4,8	3,8	Разработчик	3	2	3	2

Концептуализация системы	4	3	6	5	4,8	3,8	Разработчик	2	2	2	2
Оценка эффективности полученных результатов	1	1	2	2	1,4	1,4	Разработчик	1	1	1	1
Оценка целесообразности проведения ОКР	1	1	2	2	1,4	1,4	Разработчик	1	1	1	1
Проектирование системы и составление диаграмм	5	4	7	6	5,8	4,8	Разработчик	6	5	8	5
Оценка эффективности разработки и применения системы	1	1	2	2	1,4	1,4	Разработчик	1	1	1	1
Разработка системы	42	35	70	63	53,2	46,2	Разработчик	18	16	22	20
Тестирование разработанной системы	7	7	12	12	9	9	Разработчик	9	9	11	11
Развёртывание системы	2	2	3	3	2,4	2,4	Разработчик	3	3	3	3
Оформление эксплуатационно-технической документации	21	21	28	28	23,8	23,8	Разработчик	12	12	14	14

Составление пояснительной записки	42	42	56	56	47,6	47,6	Разработчик	16	16	20	20
---	----	----	----	----	------	------	-------------	----	----	----	----

На основе таблицы 5.10 построим диаграмму Ганта, учитывая календарную продолжительность каждого процесса. Календарный план-график представлен в таблице 5.11.

Таблица 5.11 – Календарный план-график проведения научного исследования

№ работ	Вид работ	Исполнители	Т _{ки} , кал. дн.	Продолжительность выполнения работ													
				февраль		март			апрель			май					
				2	3	1	2	3	1	2	3	1	2	3			
1	Составление и утверждение ТЗ	НР	2	■													
2	Выбор направлений исследований	НР, Р	1	■													
3	Подбор и изучение материалов по теме	Р	1		■												
4	Календарное планирование работ	НР, Р	1	■													
5	Исследование методологии построения архитектуры системы	Р	3		■												
6	Концептуализация системы	Р	2		■												
7	Оценка эффективности полученных результатов	Р	1			■											
8	Оценка целесообразности проведения ОКР	Р	1				■										
9	Проектирование системы и составление диаграмм	Р	8				■										
10	Оценка эффективности разработки и	Р	1					■									

	применения системы																		
11	Разработка системы	Р	22																
12	Тестирование разработанной системы	Р	11																
13	Развёртывание системы	Р	3																
14	Оформление эксплуатационно-технической документации	Р	14																
15	Составление пояснительной записки	Р	20																

■ – научный руководитель; ■ – разработчик

5.4. Бюджет научно-технического исследования (НТИ)

При планировании бюджета научно-технического исследования должно быть обеспечено полное и достоверное отражение всех видов расходов, связанных с его выполнением.

1. Материальные затраты.
2. Затраты на спецоборудование
3. Основная и дополнительная ЗП.
4. Социальные отчисления.
5. Прямые затраты.
6. Контрагентные расходы.
7. Накладные расходы.

5.4.1. Расчет материальных затрат

Расчет материальных затрат осуществляется по формуле:

$$Z_M = (1 + k_T) \cdot \sum_{i=1}^m C_i \cdot N_{расх\ i} , \quad (5.6)$$

где m – количество видов материальных ресурсов, потребляемых при выполнении научного исследования;

$N_{расхи}$ – количество материальных ресурсов i -го вида, планируемых к использованию при выполнении научного исследования (шт., кг, м, м² и т.д.);

$Ц_i$ – цена приобретения единицы i -го вида потребляемых материальных ресурсов (руб./шт., руб./кг, руб./м, руб./м² и т.д.);

k_T – коэффициент, учитывающий транспортно-заготовительные расходы.

Таблица 5.12 – Материальные затраты

Наименование	Единица измерения	Количество	Цена за ед., руб.	Затраты на материалы, (З _м), руб.
Электроэнергия	кВт*ч	700	5	3500
Итого, руб.				3500

Общие материальные затраты составили 3500 руб.

5.4.2. Расчет затрат на специальное оборудование для научных работ

В данную статью включают все затраты, связанные с приобретением специального оборудования (приборов, контрольно-измерительной аппаратуры, стендов, устройств и механизмов), необходимого для проведения работ по конкретной теме. Определение стоимости спецоборудования производится по действующим прейскурантам, а в ряде случаев по договорной цене. Расчет затрат по данной статье представлен в таблице 5.13.

Таблица 5.13 - Расчет бюджета затрат на приобретение спецоборудования для научных работ

Наименование	Единица измерения	Количество	Цена за ед., тыс. руб.	Затраты на материалы, (З _м), тыс. руб.
Персональный компьютер (ноутбук)	Шт.	1	70	70

Программное обеспечение (IntelliJ IDEA)	Шт.	1	0	0
Итого, тыс. руб.				70

Общие затраты на оборудование составили 70 000 руб.

5.4.3. Основная заработная плата исполнителя темы

В настоящую статью включается основная заработная плата научных и инженерно-технических работников, рабочих макетных мастерских и опытных производств, непосредственно участвующих в выполнении работ по данной теме. Величина расходов по заработной плате определяется исходя из трудоемкости выполняемых работ и действующей системы окладов и тарифных ставок. В состав основной заработной платы включается премия, выплачиваемая ежемесячно из фонда заработной платы в размере 20-30 % от тарифа или оклада. Расчет основной заработной платы приводится в таблице 5.14.

Таблица 5.14 – Расчет основной заработной платы

№ п/п	Наименование этапов	Исполнители по категориям	Трудоемкость, чел.-дн.	Зарплата, приходящаяся на один чел-дн.	Всего заработная плата по тарифу (окладам), тыс. руб.
1.	Составление и утверждение ТЗ	НР	1	5	5
2.	Выбор направлений исследований	НР, Р	1	2	2
3.	Подбор и изучение материалов по теме	Р	2	4	8
4.	Календарное планирование работ	НР, Р	1	6	6
5.	Исследование методологии построения	Р	4	4	16

	архитектуры системы				
6.	Концептуализация системы	P	4	3	12
7.	Оценка эффективности полученных результатов	P	1	3	3
8.	Оценка целесообразности проведения ОКР	P	1	2	2
9.	Проектирование системы и составление диаграмм	P	5	2	10
10.	Оценка эффективности разработки и применения системы	P	1	2	2
11	Разработка системы	P	42	5	210
12	Тестирование разработанной системы	P	7	2	14
13	Развёртывание системы	P	2	6	12
14	Оформление эксплуатационно-технической документации	P	21	3	63
15	Составление пояснительной записки	P	42	3	126
Итого:					491

Статья включает основную заработную плату работников, непосредственно занятых выполнением проекта, (включая премии, доплаты) и дополнительную заработную плату и рассчитывается по формуле:

$$Z_{зп} = Z_{осн} + Z_{доп} \quad (5.7)$$

где $Z_{\text{осн}}$ – основная заработная плата;

$Z_{\text{доп}}$ – дополнительная заработная плата (12–20 % от $Z_{\text{осн}}$).

Основная заработная плата руководителя рассчитывается по следующей формуле:

$$Z_{\text{осн}} = Z_{\text{дн}} \cdot T_p \quad (5.8)$$

где $Z_{\text{осн}}$ – основная заработная плата одного работника;

T_p – продолжительность работ, выполняемых научно-техническим работником, раб. дн.;

$Z_{\text{дн}}$ – среднедневная заработная плата работника, руб.

Среднедневная заработная плата рассчитывается по формуле:

$$Z_{\text{дн}} = \frac{Z_m \cdot M}{F_d} \quad (5.9)$$

где Z_m – месячный должностной оклад работника, руб.;

M – количество месяцев работы без отпуска в течение года:

при отпуске в 24 раб. дня $M = 11,2$ месяца, 5–дневная неделя;

при отпуске в 48 раб. дней $M = 10,4$ месяца, 6–дневная неделя;

F_d – действительный годовой фонд рабочего времени научно-технического персонала, раб. дн.

Таблица 5.15 – Баланс рабочего времени

Показатели рабочего времени	Научный руководитель	Разработчик
Календарное число дней	365	365
Количество нерабочих дней - выходные дни - праздничные дни	118	118
Потери рабочего времени - отпуск - невыходы по болезни	48 0	72 0
Действительный годовой фонд рабочего времени	199	175

Месячный должностной оклад работника (руководителя):

$$Z_{\text{м}} = Z_{\text{тс}} \cdot (1 + k_{\text{пр}} + k_{\text{д}}) \cdot k_{\text{р}} \quad (5.10)$$

где $Z_{\text{тс}}$ – заработная плата по тарифной ставке, руб.;

$k_{\text{пр}}$ – премиальный коэффициент, равный 0,3 (т.е. 30 процентов от $Z_{\text{тс}}$);

$k_{\text{д}}$ – коэффициент доплат и надбавок составляет примерно 0,2 – 0,5;

$k_{\text{р}}$ – районный коэффициент, равный 1,3 (для Томска).

Для предприятий, не относящихся к бюджетной сфере, тарифная заработная плата (оклад) рассчитывается по тарифной сетке, принятой на данном предприятии.

Расчет основной заработной платы представлен в таблице 5.16.

Таблица 5.16 – Расчет основной заработной платы

Исполнители	Разряд	$Z_{\text{тс}}$, руб.	$k_{\text{пр}}$	$k_{\text{д}}$	$k_{\text{р}}$	$Z_{\text{м}}$, руб.	$Z_{\text{дн}}$, руб.	$T_{\text{р}}$, раб. дн.	$Z_{\text{осн}}$, руб.
Научный руководитель	Старший преподаватель	30000	0,3	0,4	1,3	66300	3731,46	3	11194,37
Разработчик 1	Разработчик	15000	0,3	0,2	1,3	39000	1872,00	134	250848
Разработчик 2	Разработчик	15000	0,3	0,2	1,3	39000	1872,00	134	250848
Разработчик 3	Разработчик	15000	0,3	0,2	1,3	39000	1872,00	134	250848
Итого									763738,37

5.4.4. Расчет дополнительной заработной платы

Дополнительная заработная плата учитывает величину предусмотренных Трудовым кодексом РФ доплат за отклонение от нормальных условий труда, а также выплат, связанных с обеспечением гарантий и компенсаций (при исполнении государственных и общественных обязанностей, при совмещении работы с обучением, при предоставлении ежегодного оплачиваемого отпуска и т.д.).

Расчет дополнительной заработной платы рассчитывается по формуле:

$$Z_{\text{доп}} = k_{\text{доп}} \cdot Z_{\text{осн}}, \quad (5.11)$$

где $k_{\text{доп}}$ – коэффициент дополнительной заработной платы, принятый на стадии проектирования за 0,15.

5.4.5. Отчисления во внебюджетные фонды

В данной статье расходов отражаются обязательные отчисления по установленным законодательством Российской Федерации нормам органам государственного социального страхования (ФСС), пенсионного фонда (ПФ) и медицинского страхования (ФФОМС) от затрат на оплату труда работников.

Расчет произведен в соответствии с Федеральным законом от 24.07.2009 №212-ФЗ [11].

Так как предстоящий проект является частью сферы информационных технологий, проводим дальнейший расчет с учетом письма ФНС России от 01.03.2022 N БС-4-11/2441 [12]:

- 6% на обязательное пенсионное страхование;
- 1,5% на обязательное социальное страхование;
- 0,1% на обязательное медицинское страхование.

Таким образом общий тариф составляет 7,6%.

Отчисления во внебюджетные фонды представлены в таблице 5.17.

Таблица 5.17 – Отчисления во внебюджетные фонды

Исполнитель	Основная заработная плата, руб.			Дополнительная заработная плата, руб.		
	Исп.1	Исп.2	Исп.3	Исп.1	Исп.2	Исп.3
Руководитель проекта	11194,4	11194,4	11194,4	1679,16	1679,16	1679,16
Разработчик	250848	232128	226512	37627,2	34819,2	33976,8
Итого						
Исполнение 1	44826,62					
Исполнение 2	41554,37					
Исполнение 3	40572,69					

5.4.6. Контрагентные расходы

Контрагентные расходы включают затраты, связанные с выполнением каких-либо работ по теме сторонними организациями (контрагентами, субподрядчиками). В данном случае контрагентные расходы отсутствуют и равны 0 руб.

5.4.7. Накладные расходы

Накладные расходы учитывают прочие затраты организации, не попавшие в предыдущие статьи расходов. Их величина определяется по формуле:

$$Z_{\text{накл}} = (\sum \text{ статей}) \cdot k_{\text{нр}} \quad (5.12)$$

где $k_{\text{нр}}$ – коэффициент, учитывающий накладные расходы.

Величину коэффициента накладных расходов можно взять в размере 15%.

Накладные расходы для исполнения 1 составили:

$$Z_{\text{накл}} = (4200 + 60000 + 11194,4 + 250848 * 3 + 1679,16 + 37627,2 * 3 + 66750,73) \cdot 0,15 = 151\,397,98 \text{ руб.}$$

Накладные расходы для исполнения 2 составили:

$$Z_{\text{накл}} = (4200 + 61050 + 11194,4 + 232128 * 3 + 1679,16 + 34819,2 * 3 + 61842,35) \cdot 0,15 = 141\,208,1 \text{ руб.}$$

Накладные расходы для исполнения 3 составили:

$$Z_{\text{накл}} = (4200 + 141000 + 11194,4 + 226512 * 3 + 1679,16 + 33976,8 * 3 + 60369,83) \cdot 0,16 = 149\,959,4 \text{ руб.}$$

5.4.8. Формирование бюджета затрат научно-исследовательского проекта

Рассчитанная величина затрат научно–исследовательской работы является основой для формирования бюджета затрат проекта. Определение бюджета затрат на научно–исследовательский проект приведено в таблице 5.18.

Таблица 5.18 –Расчет бюджета затрат НИИ

Наименование статьи	Сумма, руб.			Примечание
	Исп.1	Исп.2	Исп.3	
1. Материальные затраты	4200	4200	4200	Пункт 5.4.1
2. Затраты на специальное оборудование для научных (экспериментальных) работ	60000	61050	60000	Пункт 5.4.2
3. Затраты по основной заработной плате исполнителей темы	763738,4	707578,4	690730,4	Пункт 5.4.3
4. Затраты по дополнительной заработной плате исполнителей темы	114560,76	106136,76	103609,56	Пункт 5.4.4
5. Отчисления во внебюджетные фонды	44826,62	41554,37	40572,69	Пункт 5.4.5
6. Затраты на научные и производственные командировки	-	-	-	Отсутствуют
7. Контрагентские расходы	-	-	-	Отсутствуют
8. Накладные расходы	151 397,98	141 208,1	149 959,4	Пункт 5.4.7
9. Бюджет затрат НИИ	1252907,62	1173768	1149689	

5.5. Определение ресурсной (ресурсосберегающей), финансовой, бюджетной, социальной и экономической эффективности исследования

Определение эффективности происходит на основе расчета интегрального показателя эффективности научного исследования. Его

нахождение связано с определением двух средневзвешенных величин: финансовой эффективности и ресурсоэффективности.

Интегральный показатель финансовой эффективности научного исследования определяется как:

$$I_{\text{фин.р}}^{\text{исп.}i} = \frac{\Phi_{pi}}{\Phi_{\text{max}}} \quad (5.13)$$

где $I_{\text{фин.р}}^{\text{исп.}i}$ – интегральный финансовый показатель разработки;

Φ_{pi} – стоимость i -го варианта исполнения;

Φ_{max} – максимальная стоимость исполнения научно-исследовательского проекта.

$$I_{\text{фин.р}}^{\text{исп1}} = \frac{1\,252\,907,62}{1\,252\,907,62} = 1;$$

$$I_{\text{фин.р}}^{\text{исп2}} = \frac{1\,173\,768}{1\,252\,907,62} = 0,936;$$

$$I_{\text{фин.р}}^{\text{исп3}} = \frac{1\,149\,689}{1\,252\,907,62} = 0,917.$$

Интегральный показатель ресурсоэффективности вариантов исполнения объекта исследования можно определить следующим образом:

$$I_{pi} = \sum_{i=1}^n a_i \times b_i \quad (5.14)$$

где I_{pi} – интегральный показатель ресурсоэффективности для i -го варианта исполнения разработки;

a_i – весовой коэффициент i -го варианта исполнения разработки;

b_i^a, b_i^p – балльная оценка i -го варианта исполнения разработки, устанавливается экспертным путем по выбранной шкале оценивания;

n – число параметров сравнения.

Таблица 5.19 – Сравнительная оценка характеристик вариантов исполнения проекта

Критерии	Объект исследования	Весовой коэффициент параметра	Исп.1	Исп.2	Исп.3
1. Удобство конечным пользователям		0,3	5	4	4
2. Удобство администраторам		0,3	4	4	5
3. Масштабируемость		0,1	5	5	2
4. Гибкость		0,1	5	4	2
5. Отказоустойчивость		0,2	4	4	5
Итого		1	4,5	4,1	4,1

$$I_{p-исп1} = 0,3 \cdot 5 + 0,3 \cdot 4 + 0,1 \cdot 5 + 0,1 \cdot 5 + 0,2 \cdot 4 = 4,5;$$

$$I_{p-исп2} = 0,3 \cdot 4 + 0,3 \cdot 4 + 0,1 \cdot 5 + 0,1 \cdot 4 + 0,2 \cdot 4 = 4,1;$$

$$I_{p-исп3} = 0,3 \cdot 4 + 0,3 \cdot 5 + 0,1 \cdot 2 + 0,1 \cdot 2 + 0,2 \cdot 5 = 4,3.$$

Интегральный показатель эффективности вариантов исполнения разработки ($I_{испi}$) определяется на основании интегрального показателя ресурсоэффективности и интегрального финансового показателя по формуле:

$$I_{исп1} = \frac{I_{p-исп1}}{I_{фин.р}^{исп1}} = \frac{4,5}{1} = 4,5;$$

$$I_{исп2} = \frac{I_{p-исп2}}{I_{фин.р}^{исп2}} = \frac{4,1}{0,936} = 4,38;$$

$$I_{исп3} = \frac{I_{p-исп3}}{I_{фин.р}^{исп3}} = \frac{4,1}{0,917} = 4,47.$$

Сравнение интегрального показателя эффективности вариантов исполнения разработки позволит определить сравнительную эффективность проекта и выбрать наиболее целесообразный вариант из предложенных.

Сравнительная эффективность проекта ($\mathcal{E}_{ср}$):

$$\mathcal{E}_{ср} = \frac{I_{исп2}}{I_{исп1}} \quad (5.15)$$

Таблица 5.20 – Сравнительная эффективность разработки

№	Показатели	Исп.1	Исп.2	Исп.3
1	Интегральный финансовый показатель разработки	1	0,936	0,917

2	Интегральный показатель ресурсоэффективности разработки	4,5	4,1	4,1
3	Интегральный показатель эффективности	4,5	4,38	4,47
4	Сравнительная эффективность вариантов исполнения	1	0,973	0,993

Сравнив значения интегральных показателей эффективности, можно сделать вывод, что реализация технологии в первом исполнении является более эффективным вариантом решения задачи, поставленной в данной работе с позиции финансовой и ресурсной эффективности.

5.6. Вывод по главе

В ходе выполнения раздела финансового менеджмента был проведен анализ финансово-экономических показателей разработки веб-приложения, а также был составлен список работ, продемонстрирован линейный график.

Полагаясь на результаты проведенной работы, разработка веб-приложения характеризуется как конкурентоспособная и перспективная. Имеется возможность реализации решения и в других городах Российской Федерации. Длительность непосредственной разработки составила 91 календарный день.

6. Социальная ответственность

6.1. Введение

Дипломная работа нацелена на проектирование и разработку веб-приложения для мониторинга и контроля научной и учебной исследовательской деятельности студента в научно-образовательной организации – Томском политехническом университете.

Данное веб-приложение разрабатывалось во время прохождения преддипломной практики в Томском Политехническом Университете.

Проектирование и разработка веб-приложения проходили в учебной аудитории кафедры площадью 42 квадратных метров, оборудованной мебелью для двух рабочих мест.

В данном разделе приведен анализ аспектов, оказывающих влияние на безопасность трудовой деятельности.

6.2. Правовые и организационные вопросы обеспечения безопасности

6.2.1. Правовые нормы трудового законодательства

Защита трудовых прав и свобод граждан, установление благоприятных условий труда являются целью трудового законодательства. Посредством урегулирования вопросов организации труда и управлению трудом, заработной платы порядка решения спорных ситуаций Трудовой кодекс Российской Федерации согласует интересы сторон трудовых отношений.

Наиболее важным является соблюдение следующих положений [13]:

1. Нормальная продолжительность рабочего времени не должна превышать 40 часов в неделю;
2. В течение рабочего дня (смены) работнику должен быть предоставлен

перерыв для отдыха и питания продолжительностью не более двух часов и не менее 30 минут, который в рабочее время не включается.

Правилами внутреннего трудового распорядка или трудовым договором может быть предусмотрено, что указанный перерыв может не предоставляться работнику, если установленная для него продолжительность ежедневной работы (смены) не превышает четырех часов;

3. Всем работникам предоставляются выходные дни (еженедельный непрерывный отдых).

6.2.2. Эргономические требования к правильному расположению и компоновке рабочей зоны

Рабочее место должно быть обустроено с соблюдением с учетом требований ГОСТ 12.2.032-78 «Система стандартов безопасности труда (ССБТ). Рабочее место при выполнении работ сидя. Общие эргономические требования». Должны быть соблюдены следующие условия [14]:

1. Конструкция рабочего места и взаимное расположения всех его элементов (сиденье, органы управления, средства отображения информации и т.д.) должны соответствовать антропометрическим, физиологическим и психологическим требованиям, а также характеру работы;
2. Конструкцией рабочего места должно быть обеспечено выполнение трудовых операций в пределах зоны досягаемости моторного поля.

ГОСТ Р ИСО 9241-5-2009 «Эргономические требования к проведению офисных работ с использованием видеодисплейных терминалов (VDT). Часть 5. Требования к расположению рабочей станции и осанке оператора» дополняет данный перечень тем, что конструкция мебели должна предоставлять возможность корректировки рабочих поз для выполнения нескольких задач (просмотр экрана, ввод с клавиатуры, использование

устройств неклавиатурного ввода, запись и т.д.). Так, продолжительное нахождение в статическом положении сводится к минимуму [15].

При выполнении выпускной квалификационной работы правовых и организационных нарушений по указанным требованиям не было выявлено, рабочее место было оборудовано согласно всем нормам и правилам.

6.3. Производственная безопасность

Разработка программного обеспечения веб-приложения для мониторинга и контроля научной и учебной исследовательской деятельности студента, как рабочий процесс, направленный на производство программного продукта, имеет факторы, потенциально вредные и опасные факторы для инженера на этапах проектирования и непосредственной разработки программного обеспечения.

Этот подпункт посвящен анализу факторов, влияющих на разработчика программного обеспечения на рабочем месте. Все выявленные факторы приведены в Таблице 6.1.

Таблица 6.1 - Возможные опасные и вредные производственные факторы на рабочем месте разработчика мобильного приложения

Факторы (ГОСТ 12.0.003-2015)	Нормативные документы
1. Отклонения показателей микроклимата	ГОСТ 30494-2011 «Здания жилые и общественные» [16]
2. Недостаточная освещенность рабочей зоны	СП 52.13330.2016 «Естественное и искусственное освещение» [17]
3. Монотонность труда	Трудовой кодекс Российской Федерации от 30.12.2001 №197-ФЗ (ред. от 25.02.2022) [13]

4. Опасность поражения электрическим током	ГОСТ Р 12.1.019-2017 «Система стандартов безопасности труда (ССБТ). Электробезопасность. Общие требования и номенклатура видов защиты» [18] ГОСТ 12.1.038-82 ССБТ. «Электробезопасность. Предельно допустимые уровни напряжений прикосновения и токов» [19]
--	--

6.3.1. Отклонение показателей микроклимата

Элементы, такие как температура, ее сочетание с влажностью, а также движение воздуха в рабочем пространстве, составляют микроклимат. Отклонения от принятых стандартов в отношении среды на рабочем месте, могут привести к ухудшению здоровья. Наиболее частыми причинами простудных заболеваний являются сквозняки и низкая температура, а повышенная температура может способствовать перегреву и обезвоживанию организма, постоянной усталости работника.

Рабочее помещение разработчика веб-приложения классифицируется как помещение 2-й категории для занятия умственным трудом, учебной. ГОСТ 30494-2011 «Здания жилые и общественные» [16] устанавливает оптимальные значения характеристик микроклимата на рабочем месте, которые обеспечивают благоприятные условия работы в холодный и теплый периоды года (таблица 6.2).

Таблица 6.2 - Оптимальные и допустимые нормы температуры, относительной влажности и скорости движения воздуха в обслуживаемой зоне общественных и административных зданий (категория 2) и помещений общежитий

Период года	Наименование помещения или категория	Температура воздуха, °С		Результирующая температура, °С		Относительная влажность, %		Скорость движения воздуха, м/с	
		оптимальная	допустимая	оптимальная	допустимая	оптимальная	допустимая, не	оптимальная, не	допустимая, не

							более	более	более
Холодный	2	19-21	18-23	18-20	17-22	45-30	60	0,2	0,3
Теплый	Помещения с постоянным пребыванием людей	23-25	18-28	22-24	19-27	60-30	65	0,15	0,25

Для поддержания благоприятного микроклимата должны быть использованы защитные меры, например, системы местного кондиционирования воздуха, проветривание рабочей зоны, проведение работ согласно временному регламенту и т.д.

6.3.2. Недостаточная освещенность рабочей зоны

Воздействие на зрительный аппарат оказывает освещенность рабочей зоны помещения. Значительный контраст между окружающим фоном и рабочей зоной приводит к напряжению глаз, что в свою очередь приводит к нарушениям зрительных функций и быстрой утомляемости.

СП 52.13330.2016 выделяет несколько характеристик зрительной работы в зависимости от минимального размера наблюдаемых объектов. Так, работа разработчика ПО относится к разряду зрительной работы высокой точности с наименьшим размером объекта различения 0,3-0,5 мм. В таблице 6.3 представлены требования к освещению рабочего помещения для указанного разряда.

Таблица 6.3 - Требования к освещению рабочего помещения для подразряда Б1 разряда высокой точности

Относительная продолжительность зрительной работы при направлении	Искусственное освещение				Естественное освещение
	средняя освещенность на рабочей поверхности	цилиндрическая освещенность, лк	объединенный показатель UGR, не	коэффициент пульсации освещенности и Кп, %, не	Коэффициент естественной освещенности %, при

зрения на рабочую поверхность, %	от системы общего освещения, лк, не менее		более	более	верхнем или комбинированном	боковом
Не менее 70	400	100	19	15	3,0	1,0

К снижению факторы недостаточной освещенности способствуют меры такие, как правильно спроектированное и рационально выполненное освещение производственных помещений. Главной целью является снижение контраста яркости экрана, окружающего естественного освещения с оконных проемов и искусственного освещения, которым оборудовано помещение. Помимо этого, каждому сотруднику необходимо придерживаться рациональному режиму труда и отдыха.

6.3.3. Монотонность труда

Зачастую работа разработчика ПО отличается однообразием трудовых процессов и окружающей производственной среды, что выделяет данный вид деятельности ее монотонностью. Сотрудник вследствие продолжительной монотонной работы сотрудник может испытывать негативные влияния такие, как умственное напряжение, утомляемость, сонливость и т.д.

Средством избежание монотонности труда является разбавление рабочего процесса различными активностями:

1. Регулярные перерывы во время рабочего дня;
2. Регулярные фазы сна и работы в течение суток;
3. Занятие спортом вне рабочего времени;

Данные советы в первую очередь направлены на расслабление нервной системы и отвлечение человека от чрезмерной концентрации на одной и той активности.

6.3.4. Опасность поражения электрическим током

Непосредственная близость к электрическим приборам и сетям, систем электрического питания является повсеместным опасным фактором в деятельности для разработчика ПО. Поражение электрическим током может

случиться по причине некачественной электропроводки работы с неисправными аппаратами, компьютерной техники и т.д.

Поражение электрическим током, которое наиболее часто проявляется в виде ожогов, судорожного сокращения мышц, происходит в результате соприкосновения человека с находящимися под напряжением токоведущими частями [18]. В зависимости от силы воздействия, в некоторых случаях подобные инциденты могут привести к потере сознания, нарушениям деятельности внутренних органов и гибели людей.

Правила ГОСТ 12.1.038-82 ССБТ. «Электробезопасность. Предельно допустимые уровни напряжений прикосновения и токов» обобщают те значения напряжения и силы тока, протекающих через тело человека при нормальном (неаварийном) режиме электроустановки, превышение которых недопустимо (табл. 6.4) [19].

Таблица 6.4 - Максимальные значения напряжения и силы тока, протекающих через тело человека при нормальном режиме электроустановки

Род тока	Напряжение, В	I, мА
	не более	
Переменный, 50 Гц	2,0	0,3
Переменный, 400 Гц	3,0	0,4
Постоянный	8,0	1,0

Для защиты от воздействий электрического тока необходимо минимизировать доступность приборов, находящихся под напряжением, и предотвратить их случайное соприкосновение с человеческим телом. Так, изолирующие и оградительные устройства используются в непосредственной близости от рабочей зоны с соответствующей предупреждающей надписью.

Также используются следующие меры для предупреждения подобных инцидентов:

- проведение регулярных инструктажей техники безопасности;
- исключение наличия посторонних предметов, еды, напитков на

рабочем месте, оборудованном ПК;

- проведение быстрого осмотра исправности приборов перед началом трудовой активности;
- оповещение ответственного лица об обнаруженных неисправностях.

6.4. Экологическая безопасность

Разработка веб-приложения для мониторинга и контроля исследовательской деятельности не требовала использования более технологических устройств, чем персональный компьютер, ноутбук, принтер и смартфон. Но несмотря на то, что их эксплуатация приносит незначительный вред, их утилизация является глобальной проблемой.

Неправильно утилизированная оргтехника оказывает огромный вред на литосферу. Выброшенные комплектующие ЭВМ, например, батарейки, могут содержать в себе вредные химические элементы, наносящие ущерб почве, делая ее непригодной для дальнейшего ее использования. Более того, те же токсичные вещества, содержащиеся в компьютерах, могут напрямую контактировать с водой в реках и водоемах либо проникнуть глубже в недра Земли, вызывая дальнейшее загрязнение подземных вод.

Утилизация деталей ЭВМ путем сжигания является не менее вредным фактором. Многие из них сделаны из пластика, сгорание которого загрязняет воздух, что в больших масштабах влияет на климат.

Согласно ГОСТ Р 55102-2012 «Ресурсосбережение. Обращение с отходами. Руководство по безопасному сбору, хранению, транспортированию и разборке отработавшего электротехнического и электронного оборудования, за исключением ртутьсодержащих устройств и приборов» сбор отработавших электротехнического и электронного оборудования производится в местах, где осуществляется распространение электрического или электронного оборудования среди конечных потребителей [20]. После этого различные виды ОЭЭО собираются в отдельные упаковки и направляются для повторного внедрения.

Путем правильного соблюдения мероприятий по эксплуатации и утилизации персональных компьютеров и ноутбуков, а также их комплектующих каждый сотрудник может внести вклад в сохранение экологической безопасности.

6.5. Безопасность в чрезвычайных ситуациях

При разработке программного обеспечения пожар является наиболее вероятной чрезвычайной ситуацией. Очагом пожара могут послужить такие факторы как короткое замыкание, неисправность или чрезмерная перегрузка оборудования, которые могут вызвать нагрев легковоспламеняющихся предметов как бумага, мебель и другие.

Наиболее частыми причинами возникновения пожара являются короткое замыкание, неисправность оборудования, перегрузка сетей с нагревом токоведущих частей. На основании Федерального закона от 22.07.2008 N 123-ФЗ [22] в зависимости от горючих объектов пожар наиболее вероятно будет соответствовать классам А или Е.

Среди первичных средств пожаротушения рабочее место оборудовано порошковыми огнетушителями, предназначенные для устранения возгорания типа АВСЕ. Дополнительно к мероприятиям противопожарного режима относится соблюдение следующих правил противопожарного режима [21]:

1. Огнетушители должны всегда содержаться в исправном состоянии, периодически осматриваться, проверяться и своевременно перезаряжаться;
2. Расстояние от возможного очага пожара до места размещения огнетушителя не должно превышать 20 м для общественных зданий и сооружений;
3. Огнетушители, размещенные в коридорах, проходах, не должны препятствовать безопасной эвакуации людей. Огнетушители следует располагать на видных местах вблизи от выходов из помещений на высоте не более 1,5 метра до верха корпуса огнетушителя;

Для предотвращения возникновения пожара необходимо:

1. регулярно проводить инструктажи сотрудников предприятия по пожарной безопасности;
2. разместить в помещении план эвакуации и плакаты с краткой информацией с действиями при возникновении пожара;
3. оборудовать помещение пожарной сигнализацией и красными кнопками, а также средствами тушения пожара.

6.6. Выводы по разделу

В ходе выполнения раздела социальная ответственность был проведен анализ факторов, оказывающих влияние на безопасность трудовой деятельности программиста по разработке мобильного приложения, а также возможных опасных и вредных факторов, влияющие на производственную, экологическую безопасности и безопасность в чрезвычайных ситуациях.

Работа по проектированию и разработке мобильного приложения в соответствии с СП 52.13330.2016 характеризуется разрядом зрительной работы высокой точности и требует особого внимания к соблюдению требований освещенности. Работа проводилась в аудиториях Кибернетического центра и учебной аудитории кафедры. По взрывопожарной и пожарной опасности оба помещения относятся к категории Г умеренной пожароопасности, а по электробезопасности характеризуются как помещения без повышенной опасности поражения электрическим током с квалификационной группой по электробезопасности персонала - I. Рабочее помещение разработчика относится к помещениям 2-й категории, предназначенным для людей, занятых умственным трудом, учебной.

Для экологии эксплуатация вычислительных машин несет малый урон по сравнению с их неправильной утилизацией, токсичные продукты которой вызывают серьезные проблемы на уровнях литосферы, гидросферы и атмосферы. С другой стороны, при правильной утилизации комплектующие могут быть повторно использованы и внедрены в новые разработки.

Заключение

В ходе проделанной работы были проанализированы глобальные этапы исследовательской работы студента и определены:

1. Основная проблема: неосведомленность, несопоставимость интересов студента и руководителя.
2. Основная цель: обеспечение педагогов и руководства вуза удобством понимания и визуализации ситуации проектной деятельности как отдельного студента, так и руководителей.
3. Тип приложения: веб-приложение.
4. Основным инструментом для серверной части приложения: Java и Spring Boot.
5. СУБД: MySQL.

Основная проблема была решена, а цель – достигнута с помощью спроектированной и разработанной информационной системы мониторинга и контроля учебной и научной исследовательской деятельности.

Главными чертами программы стали интегрируемость и востребованность.

Тема актуальна благодаря пользе приложения как средства, порождающего правила взаимоотношений между преподавателями и студентами университета или влияющего на них. Роль такой системы будет определяться улучшением качественных признаков, которые воздействуют на будущих выпускников со стороны учебного заведения на протяжении всего времени их обучения.

Исходный код можно найти по ссылке на репозиторий Github (приложение К).

Список источников

1. MySQL vs Oracle:6 Comprehensive Comparisons : сайт. – URL: <https://hevodata.com/learn/mysql-vs-oracle/#Key1> (дата обращения: 15.03.2022). – Текст : электронный.
2. Шерстнёв В.С., Распопов А.В. Разработка интернет-каталога для организации доступа к корпоративному хранилищу электронных документов ТПУ // Современные проблемы науки и образования. – 2013. – № 6. ; URL: <https://science-education.ru/ru/article/view?id=11130> (дата обращения: 15.03.2022). – Текст : электронный.
3. Корпоративный портал ТПУ : сайт. – URL: <https://portal.tpu.ru/portal/page/portal/www> (дата обращения: 15.05.2022). – Текст : электронный.
4. Spring Boot // Spring : сайт. – URL: <https://spring.io/projects/spring-boot> (дата обращения: 01.04.2022). – Текст : электронный.
5. Особенности Spring Boot // Runebook.dev : сайт. – URL: https://runebook.dev/ru/docs/spring_boot/spring-boot-features (дата обращения: 03.04.2022). – Текст : электронный.
6. Руководство по аннотациям Spring Framework // CoderLessons.com : сайт. – URL: <https://coderlessons.com/articles/java/rukovodstvo-po-annotatsiiam-spring-framework> (дата обращения: 05.04.2022). – Текст : электронный.
7. Шпаргалка Java программиста 1: JPA и Hibernate в вопросах и ответах // Хабр : сайт. – URL: <https://habr.com/ru/post/265061/> (дата обращения: 05.03.2022). – Текст : электронный.
8. Hibernate. Everything data // Hibernate : сайт. – URL: <https://hibernate.org/> (дата обращения: 05.04.2022). – Текст : электронный.

9. Derived Query Methods in Spring Data JPA Repositories // Baeldung : сайт.
– URL: <https://www.baeldung.com/spring-data-derived-queries> (дата обращения: 07.04.2022). – Текст : электронный.
10. How to use the atomic compare and swap technique in Java // educative.io : сайт. – URL: <https://www.educative.io/edpresso/how-to-use-the-atomic-compare-and-swap-technique-in-java> (дата обращения: 10.04.2022). – Текст : электронный.
11. Закон Российской Федерации «Федеральный закон №212-ФЗ» от 24.07.2009 // Российская газета.
12. Закон Российской Федерации «письмо ФНС России» от 01.03.2022 № N БС-4-11/2441 // Российская газета.
13. Трудовой кодекс Российской Федерации от 30.12.2001 N 197-ФЗ (ред. от 25.02.2022) (с изм. и доп., вступ. в силу с 01.03.2022). – Текст : электронный // КонсультантПлюс : [сайт]. – URL: http://www.consultant.ru/document/cons_doc_law_34683/ (дата обращения: 20.05.2022).
14. ГОСТ 12.2.032-78 Система стандартов безопасности труда (ССБТ). Рабочее место при выполнении работ сидя. Общие эргономические требования: дата введения 1979-01-01. – URL: <https://docs.cntd.ru/document/1200003913> (дата обращения: 20.05.2022). – Текст : электронный.
15. СанПиН 2.2.2/2.4.1340-03 Гигиенические требования к персональным электронно-вычислительным машинам и организации работы (с изменениями от 25 апреля 2007 г.) – URL: https://ekosf.ru/wp-content/uploads/2020/06/sanpin_2.2-2.4.1340-03.pdf (дата обращения: 20.05.2022). – Текст : электронный.
16. СанПиН 2.2.4.548-96 Гигиенические требования к микроклимату производственных помещений. – URL:

- <https://docs.cntd.ru/document/901704046> (дата обращения: 20.05.2022). – Текст : электронный.
- 17.СП 52.13330.2016 Естественное и искусственное освещение: дата введения 2017-05-08. – URL: <https://docs.cntd.ru/document/456054197> (дата обращения: 20.05.2022). – Текст : электронный.
- 18.СН 2.2.4/2.1.8.562-96 Шум на рабочих местах, в помещениях жилых, общественных зданий и на территории жилой застройки. – URL: <https://docs.cntd.ru/document/901703278> (дата обращения: 20.05.2022). – Текст : электронный.
- 19.ГОСТ Р 12.1.019-2017 «Система стандартов безопасности труда (ССБТ). Электробезопасность. Общие требования и номенклатура видов защиты» [..]
- 20.ГОСТ 12.1.038-82 ССБТ. «Электробезопасность. Предельно допустимые уровни напряжений прикосновения и токов».
- 21.Меры защиты от случайного поражения электрическим током. – URL: <http://okhrana-truda.com/elektrobezopasnost/mery-zashchity-ot-sluchajnogo-porazheniya-elektricheskim-tokom.html> (дата обращения: 20.05.2022). – Текст : электронный.
- 22.ГОСТ Р 55090-2012 Ресурсосбережение. Обращение с отходами. Рекомендации по утилизации отходов бумаги: дата введения 2014-01-01. – URL: <https://docs.cntd.ru/document/1200103182> (дата обращения: 20.05.2022). – Текст : электронный.

Приложение А. Код DDL базы данных

```
SET @OLD_UNIQUE_CHECKS=@@UNIQUE_CHECKS, UNIQUE_CHECKS=0;
SET @OLD_FOREIGN_KEY_CHECKS=@@FOREIGN_KEY_CHECKS, FOREIGN_KEY_CHECKS=0;
SET @OLD_SQL_MODE=@@SQL_MODE,
SQL_MODE='ONLY_FULL_GROUP_BY,STRICT_TRANS_TABLES,NO_ZERO_IN_DATE,NO_ZERO_DATE
,ERROR_FOR_DIVISION_BY_ZERO,NO_ENGINE_SUBSTITUTION';
```

```
-----
-- Schema mydb
-----
```

```
CREATE SCHEMA IF NOT EXISTS `mydb` DEFAULT CHARACTER SET utf8 ;
USE `mydb` ;
```

```
-----
-- Table `mydb`.`user`
-----
```

```
CREATE TABLE `mydb`.`user` (
  `email` VARCHAR(20) NOT NULL,
  `password` VARCHAR(45) NOT NULL,
  PRIMARY KEY (`email`));
```

```
-----
-- Table `mydb`.`school`
-----
```

```
CREATE TABLE IF NOT EXISTS `mydb`.`school` (
  `school_id` INT AUTO_INCREMENT NOT NULL,
  `name` VARCHAR(256) NOT NULL,
  PRIMARY KEY (`school_id`));
```

```
-----
-- Table `mydb`.`department`
-----
```

```
CREATE TABLE IF NOT EXISTS `mydb`.`department` (
  `department_id` INT AUTO_INCREMENT NOT NULL,
  `name` VARCHAR(256) NOT NULL,
  `school_id` INT NOT NULL,
  PRIMARY KEY (`department_id`),
  INDEX `fk_department` (`school_id` ASC) VISIBLE,
  CONSTRAINT `fk_department`
  FOREIGN KEY (`school_id`)
  REFERENCES `mydb`.`school` (`school_id`));
```

```
-----
-- Table `mydb`.`speciality`
-----
```

```
CREATE TABLE IF NOT EXISTS `mydb`.`speciality` (
  `speciality_id` INT AUTO_INCREMENT NOT NULL,
  `name` VARCHAR(128) NOT NULL,
  `department_id` INT NOT NULL,
  PRIMARY KEY (`speciality_id`),
  INDEX `fk_speciality` (`department_id` ASC) VISIBLE,
  CONSTRAINT `fk_speciality`
  FOREIGN KEY (`department_id`)
  REFERENCES `mydb`.`department` (`department_id`));
```

```
-----
-- Table `mydb`.`group_`
-----
```

```

-----
CREATE TABLE IF NOT EXISTS `mydb`.`group_` (
  `group_id` INT AUTO_INCREMENT NOT NULL,
  `name` VARCHAR(128) NOT NULL,
  `enrollment_year` INT NOT NULL,
  `course` INT NOT NULL,
  `speciality_id` INT NOT NULL,
  PRIMARY KEY (`group_id`),
  INDEX `fk_group` (`speciality_id` ASC) VISIBLE,
  CONSTRAINT `fk_group`
    FOREIGN KEY (`speciality_id`)
    REFERENCES `mydb`.`speciality` (`speciality_id`));
-----

```

```

-- Table `mydb`.`student`
-----

```

```

CREATE TABLE IF NOT EXISTS `mydb`.`student` (
  `student_id` INT AUTO_INCREMENT NOT NULL,
  `last_name` VARCHAR(128) NOT NULL,
  `name` VARCHAR(128) NOT NULL,
  `patronymic` VARCHAR(128) NOT NULL,
  `email` VARCHAR(512) NOT NULL,
  `personal_id` INT NOT NULL,
  `group_id` INT NOT NULL,
  PRIMARY KEY (`student_id`),
  INDEX `fk_student` (`group_id` ASC) VISIBLE,
  INDEX `fk_user_student_idx` (`email` ASC) VISIBLE,
  CONSTRAINT `fk_user_student`
    FOREIGN KEY (`email`)
    REFERENCES `mydb`.`user` (`email`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION,
  CONSTRAINT `fk_student`
    FOREIGN KEY (`group_id`)
    REFERENCES `mydb`.`group_` (`group_id`));
-----

```

```

-- Table `mydb`.`academic_position`
-----

```

```

CREATE TABLE IF NOT EXISTS `mydb`.`academic_position` (
  `acad_pos_id` INT AUTO_INCREMENT NOT NULL,
  `name` VARCHAR(128) NOT NULL,
  PRIMARY KEY (`acad_pos_id`));
-----

```

```

-- Table `mydb`.`science_degree`
-----

```

```

CREATE TABLE IF NOT EXISTS `mydb`.`science_degree` (
  `degree_id` INT AUTO_INCREMENT NOT NULL,
  `name` VARCHAR(128) NOT NULL,
  PRIMARY KEY (`degree_id`));
-----

```

```

-- Table `mydb`.`teacher`
-----

```

```

CREATE TABLE IF NOT EXISTS `mydb`.`teacher` (
  `teacher_id` INT AUTO_INCREMENT NOT NULL,
  `last_name` VARCHAR(128) NOT NULL,
-----

```

```

`name` VARCHAR(128) NOT NULL,
`patronymic` VARCHAR(128) NOT NULL,
`email` VARCHAR(128) NOT NULL,
`personal_id` INT NOT NULL,
`acad_pos_id` INT NOT NULL,
`department_id` INT NOT NULL,
`degree_id` INT NOT NULL,
PRIMARY KEY (`teacher_id`),
INDEX `fk_teacher_acad_pos` (`acad_pos_id` ASC) VISIBLE,
INDEX `fk_teacher_department` (`department_id` ASC) VISIBLE,
INDEX `fk_teacher_degree` (`degree_id` ASC) VISIBLE,
INDEX `fk_user_teacher_idx` (`email` ASC) VISIBLE,
CONSTRAINT `fk_teacher_acad_pos`
  FOREIGN KEY (`acad_pos_id`)
  REFERENCES `mydb`.`academic_position` (`acad_pos_id`),
CONSTRAINT `fk_teacher_department`
  FOREIGN KEY (`department_id`)
  REFERENCES `mydb`.`department` (`department_id`),
CONSTRAINT `fk_teacher_degree`
  FOREIGN KEY (`degree_id`)
  REFERENCES `mydb`.`science_degree` (`degree_id`),
CONSTRAINT `fk_user_teacher`
  FOREIGN KEY (`email`)
  REFERENCES `mydb`.`user` (`email`)
  ON DELETE NO ACTION
  ON UPDATE NO ACTION);

```

```

-----
-- Table `mydb`.`scientific_work`
-----

```

```

CREATE TABLE IF NOT EXISTS `mydb`.`scientific_work` (
  `work_id` INT AUTO_INCREMENT NOT NULL,
  `title` VARCHAR(512) NOT NULL,
  `teacher_id` INT NOT NULL,
  PRIMARY KEY (`work_id`),
  INDEX `fk_scientific_work` (`teacher_id` ASC) VISIBLE,
  CONSTRAINT `fk_scientific_work`
    FOREIGN KEY (`teacher_id`)
    REFERENCES `mydb`.`teacher` (`teacher_id`));

```

```

-----
-- Table `mydb`.`student_work_type`
-----

```

```

CREATE TABLE IF NOT EXISTS `mydb`.`student_work_type` (
  `type` VARCHAR(128) NOT NULL,
  `name` VARCHAR(128) NOT NULL,
  PRIMARY KEY (`type`));

```

```

-----
-- Table `mydb`.`student_work`
-----

```

```

CREATE TABLE IF NOT EXISTS `mydb`.`student_work` (
  `student_id` INT NOT NULL,
  `work_id` INT NOT NULL,
  `title` VARCHAR(512) NOT NULL,
  `start_date` DATE NULL DEFAULT NULL,
  `completion_date` DATE NULL DEFAULT NULL,
  `score` INT DEFAULT NULL,
  `leave_reason` VARCHAR(128) DEFAULT NULL,
  `type` VARCHAR(128) NOT NULL,

```

```

PRIMARY KEY (`student_id`, `work_id`),
INDEX `fk_student_work_2` (`work_id` ASC) VISIBLE,
INDEX `fk_student_work_student_work_type1_idx` (`type` ASC) VISIBLE,
CONSTRAINT `fk_student_work_1`
  FOREIGN KEY (`student_id`)
  REFERENCES `mydb`.`student` (`student_id`),
CONSTRAINT `fk_student_work_2`
  FOREIGN KEY (`work_id`)
  REFERENCES `mydb`.`scientific_work` (`work_id`),
CONSTRAINT `fk_student_work_student_work_type1`
  FOREIGN KEY (`type`)
  REFERENCES `mydb`.`student_work_type` (`type`)
ON DELETE NO ACTION
ON UPDATE NO ACTION);

```

```

-----
-- Table `mydb`.`science_specialization`
-----

```

```

CREATE TABLE `science_specialization` (
  `specialization_id` int NOT NULL AUTO_INCREMENT,
  `name` varchar(128) NOT NULL,
  PRIMARY KEY (`specialization_id`)
);

```

```

-----
-- Table `mydb`.`teacher_specialization`
-----

```

```

CREATE TABLE `teacher_specialization` (
  `teacher_id` int NOT NULL,
  `specialization_id` int NOT NULL,
  PRIMARY KEY (`teacher_id`, `specialization_id`),
  INDEX `fk_specialization_idx` (`specialization_id` ASC) VISIBLE,
  INDEX `fk_teacher_idx` (`teacher_id` ASC) VISIBLE,
  CONSTRAINT `fk_specialization` FOREIGN KEY (`specialization_id`)
  REFERENCES `science_specialization` (`specialization_id`),
  CONSTRAINT `fk_teacher` FOREIGN KEY (`teacher_id`) REFERENCES `teacher`
  (`teacher_id`)
);

```

```

-----
-- Table `mydb`.`role`
-----

```

```

CREATE TABLE `mydb`.`role` (
  `role_id` int NOT NULL,
  `name` varchar(45) NOT NULL,
  PRIMARY KEY (`role_id`));

```

```

-----
-- Table `mydb`.`user_role`
-----

```

```

CREATE TABLE `mydb`.`user_role` (
  `email` varchar(45) NOT NULL,
  `role_id` int NOT NULL,
  PRIMARY KEY (`email`, `role_id`),
  INDEX `fk_role_idx` (`role_id` ASC) INVISIBLE,
  INDEX `fk_email_idx` (`email` ASC) VISIBLE,
  CONSTRAINT `fk_email`
  FOREIGN KEY (`email`)

```

```

        REFERENCES `mydb`.`user` (`email`)
        ON DELETE NO ACTION
        ON UPDATE NO ACTION,
CONSTRAINT `fk_role`
        FOREIGN KEY (`role_id`)
        REFERENCES `mydb`.`role` (`role_id`)
        ON DELETE NO ACTION
        ON UPDATE NO ACTION);
SET SQL_MODE=@OLD_SQL_MODE;
SET FOREIGN_KEY_CHECKS=@OLD_FOREIGN_KEY_CHECKS;
SET UNIQUE_CHECKS=@OLD_UNIQUE_CHECKS;

```

```

-----
-- Table `mydb`.`review`
-----

```

```

CREATE TABLE `mydb`.`review` (
  `review_id` INT NOT NULL,
  `from_email` VARCHAR(20) NOT NULL,
  `of_email` VARCHAR(20) NOT NULL,
  `comment` VARCHAR(45) NOT NULL,
  `creation_date` DATE NOT NULL,
  INDEX `fk_from_idx` (`from_email` ASC) VISIBLE,
  INDEX `fk_of_idx` (`of_email` ASC) VISIBLE,
  CONSTRAINT `fk_from`
    FOREIGN KEY (`from_email`)
      REFERENCES `mydb`.`user` (`email`)
      ON DELETE NO ACTION
      ON UPDATE NO ACTION,
  CONSTRAINT `fk_of`
    FOREIGN KEY (`of_email`)
      REFERENCES `mydb`.`user` (`email`)
      ON DELETE NO ACTION
      ON UPDATE NO ACTION,
  PRIMARY KEY (`review_id`));

```

Приложение Б. Настройка источника данных и JPA

```
spring.datasource.url=jdbc:mysql://localhost:3306/mydb
spring.datasource.username=root
spring.datasource.password=fdmdh
spring.datasource.driver-class-name=com.mysql.cj.jdbc.Driver
spring.sql.init.mode=always
```

```
spring.jpa.hibernate.ddl-auto=update
spring.jpa.database-
platform=org.hibernate.dialect.MySQL5InnoDBDialect
spring.jpa.show-sql=true
```

Приложение В. Реализация JPA репозитория для сущности Student

```
public interface StudentRepository extends JpaRepository<Student, Long> {
    @Query("select s from Student s where s.email = :email")
    Student findByEmail(@Param("email") String email);
    @Query("select s from Student s where s.personal_id = ?1")
    Student findByPersonal_id(Long personalId);
    @Query("select s from Student s " +
        "where upper(s.last_name) like upper(concat('%', :input, '%')) or
        upper(s.name) like upper(concat('%', :input, '%')) or upper(s.patronymic)
        like upper(concat('%', :input, '%'))")
    List<Student> customFindByInput(@Param("input") String input);
    @Query("select s from Student s where s.student_id = :id")
    Student findFirstByStudent_id(@Param("id") Long id);
}

@Service
public class StudentService {
    private StudentRepository studentRepository;
    @Autowired
    public StudentService(StudentRepository studentRepository) {
        this.studentRepository = studentRepository;
    }
    public List<Student> findAll() {
        return studentRepository.findAll();
    }
    public List<Student> findByInput(String input) {
        return studentRepository.customFindByInput(input);
    }
    public Student findById(Long id) {
        return studentRepository.findFirstByStudent_id(id);
    }
}
```

Приложение Г. Настройка Entity классов для таблиц Teacher, Science_specialization и Teacher_specialization

```
@ToString
@Getter
@Setter
@Entity
@NoArgsConstructor
public class Teacher extends Person {

    @Id
    @GeneratedValue(strategy = GenerationType.IDENTITY)
    private Long teacher_id;
    private String last_name;
    private String name;
    private String patronymic;
    private String email;
    @Column(unique=true)
    private Long personal_id;

    @ManyToOne
    @JoinColumn(name = "acad_pos_id")
    private Academic_position academicPosition;

    @ManyToOne
    @JoinColumn(name = "department_id")
    private Department department;

    @ManyToOne
    @JoinColumn(name = "degree_id")
    private Science_degree degree;

    @ManyToMany
    @JoinTable(
        name = "teacher_specialization",
        joinColumns = @JoinColumn(
            name = "teacher_id",
            referencedColumnName = "teacher_id"
        ),
        inverseJoinColumns = @JoinColumn(
            name = "specialization_id",
            referencedColumnName = "specialization_id"
        )
    )
    @ToString.Exclude
    private List<Science_specialization>
specializationList;

    @Override
    public boolean equals(Object o) {
        if (this == o) return true;
        if (o == null || Hibernate.getClass(this) !=
Hibernate.getClass(o)) return false;
        Teacher teacher = (Teacher) o;
        return teacher_id != null &&
Objects.equals(teacher_id, teacher.teacher_id);
    }

    @Override
    public int hashCode() {
        return getClass().hashCode();
    }
}
```

```

    }

    public static Comparator<Teacher>comparatorTeacher()
    {
        return Comparator
            .comparing(Teacher::getLast_name)
            .thenComparing(Teacher::getName)
            .thenComparing(Teacher::getPatronymic);
    }
}

@Entity
@ToString
@Getter
@Setter
@NoArgsConstructor
public class Teacher_specialization {

    @EmbeddedId
    private MyTeacherSpecializationKey key;
}

@Embeddable
@ToString
@Getter
@Setter
@NoArgsConstructor
@EqualsAndHashCode
public class MyTeacherSpecializationKey implements Serializable {

    private Long teacher_id;
    private Long specialization_id;
}

@Entity
@ToString
@Getter
@Setter
@NoArgsConstructor
public class Science_specialization {

    @Id
    @GeneratedValue(strategy = GenerationType.IDENTITY)
    private Long specialization_id;
    private String name;
    @ManyToMany
    @JoinTable(name = "teacher_specialization",
        joinColumns = @JoinColumn(
            name = "specialization_id",
            referencedColumnName = "specialization_id"
        ),
        inverseJoinColumns = @JoinColumn(
            name = "teacher_id",
            referencedColumnName = "teacher_id"
        )
    )
    @ToString.Exclude
    private List<Teacher> teacherList;
    @Override
    public boolean equals(Object o) {
        if (this == o) return true;
        if (o == null || Hibernate.getClass(this) != Hibernate.getClass(o))
return false;

```

```

    Science_specialization that = (Science_specialization) o;
    return specialization_id != null && Objects.equals(specialization_id,
that.specialization_id);
}

@Override
public int hashCode() {
    return getClass().hashCode();
}
}

```

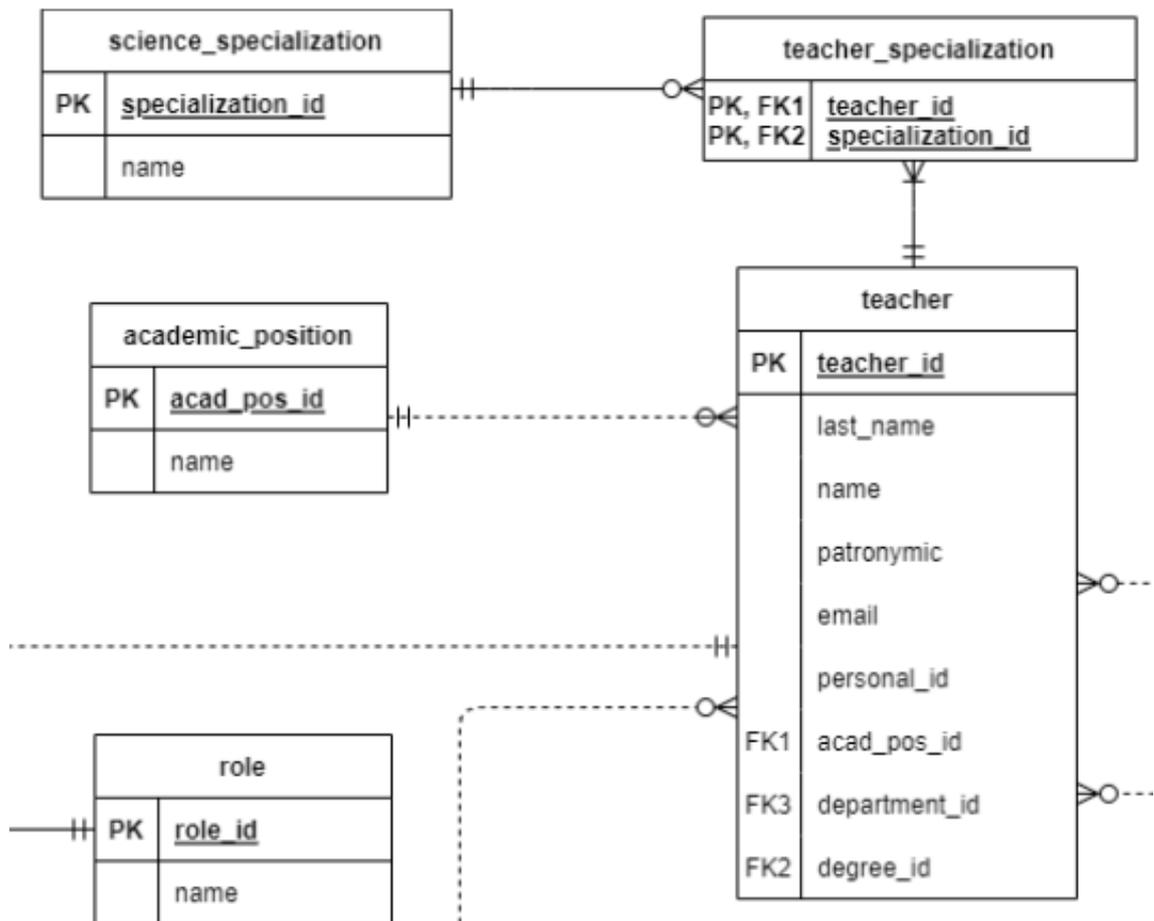


Рисунок I – Логическая модель данных для таблиц Teacher, Science_specialization и Teacher_specialization

Приложение Д. Настройка доступа к ресурсам веб-приложения

```
@Configuration
@EnableWebSecurity
public class WebSecurityConfig extends WebSecurityConfigurerAdapter {
    @Autowired
    LoginService userService;

    @Bean
    public BCryptPasswordEncoder bCryptPasswordEncoder() {
        return new BCryptPasswordEncoder();
    }

    @Override
    protected void configure(HttpSecurity httpSecurity) throws Exception {
        httpSecurity
            .csrf()
            .disable()
            .authorizeRequests()
            .antMatchers("/profile/**").fullyAuthenticated()
            .anyRequest().permitAll()
            .and()
            .formLogin()
            .loginPage("/sso")
            .defaultSuccessUrl("/main")
            .permitAll()
            .and()
            .logout()
            .permitAll()
            .logoutSuccessUrl("/");
    }

    @Autowired
    protected void configureGlobal(AuthenticationManagerBuilder auth) throws
Exception {
        auth.userDetailsService(userService).passwordEncoder(bCryptPasswordEnc
oder());
    }
}
```

Приложение Е. Код Контроллера для страницы «Все работы»

```
@Controller
public class WorksController {
```

```

private final WorksService worksService;

@Autowired
public WorksController(WorksService worksService) {
    this.worksService = worksService;
}

@ModelAttribute("teacherList")
public List<Teacher> teacherList() {
    return worksService.findAllTeachers()
        .stream()
        .sorted(Teacher.comparatorTeacher()).toList();
}

@ModelAttribute("studentList")
public List<Student> studentList() {
    return worksService.findAllStudents()
        .stream()
        .sorted(Student.comparatorStudent()).toList();
}

@ModelAttribute("newStudentList")
public List<Student> newStudentList() {
    return new ArrayList<>();
}

@GetMapping("/works")
public String worksPage(@NotNull Model model) {
    worksService.findAllWorks().keySet().forEach(s ->
Logger.getLogger(this.getClass().getName()).info(s.toString()));

    val studentForWorkMap =
worksService.findAllWorks().keySet().stream().toList().stream().collect(Colle
ctors.toMap(s -> s, s -> worksService.findStudentsByWork_id(s.getWork_id()),
(a, b) -> b));
    model.addAttribute("workList",
worksService.findAllWorks().values().stream().toList())
        .addAttribute("scientificWorkList",
worksService.findAllWorks().keySet().stream().toList())
        .addAttribute("studentForWorkMap", studentForWorkMap)
        .addAttribute("tempWork", new Student_work());

    return "student_works";
}

@GetMapping("/works/filter/teacher/{id}")
public String filterTeacher(@PathVariable("id") Long id, @NotNull Model
model) {
    worksService.findWorksByTeacher_id(id).keySet().forEach(s ->
Logger.getLogger(this.getClass().getName()).info(s.toString()));

    val studentForWorkMap =
worksService.findWorksByTeacher_id(id).keySet().stream().toList().stream().co
llect(Collectors.toMap(s -> s, s ->
worksService.findStudentsByWork_id(s.getWork_id()), (a, b) -> b));
    model.addAttribute("workList",
worksService.findWorksByTeacher_id(id).values().stream().toList())
        .addAttribute("scientificWorkList",
worksService.findWorksByTeacher_id(id).keySet().stream().toList())
        .addAttribute("studentForWorkMap", studentForWorkMap)
        .addAttribute("tempWork", new Student_work());
}

```

```

        return "student_works";
    }

    @GetMapping("/works/filter/student/{id}")
    public String filterStudent(@PathVariable("id") Long id, @NotNull Model
model) {
        worksService.findWorksByStudent_id(id).keySet().forEach(s ->
Logger.getLogger(this.getClass().getName()).info(s.toString()));

        val studentForWorkMap =
worksService.findWorksByStudent_id(id).keySet().stream().toList().stream().co
llect(Collectors.toMap(s -> s, s ->
worksService.findStudentsByWork_id(s.getWork_id()), (a, b) -> b));
        model.addAttribute("workList",
worksService.findWorksByStudent_id(id).values().stream().toList()
            .addAttribute("scientificWorkList",
worksService.findWorksByStudent_id(id).keySet().stream().toList()
            .addAttribute("studentForWorkMap", studentForWorkMap)
            .addAttribute("tempWork", new Student_work()));

        return "student_works";
    }

    @PostMapping("/works/filter/title")
    public String filterTitle(@ModelAttribute("tempWork") @NotNull
Student_work tempWork, @NotNull Model model) {
        worksService.findByTitle(tempWork.getTitle()).keySet().forEach(s ->
Logger.getLogger(this.getClass().getName()).info(s.toString()));

        val studentForWorkMap =
worksService.findByTitle(tempWork.getTitle()).keySet().stream().toList().stre
am().collect(Collectors.toMap(s -> s, s ->
worksService.findStudentsByWork_id(s.getWork_id()), (a, b) -> b));
        model.addAttribute("workList",
worksService.findByTitle(tempWork.getTitle()).values().stream().toList()
            .addAttribute("scientificWorkList",
worksService.findByTitle(tempWork.getTitle()).keySet().stream().toList()
            .addAttribute("studentForWorkMap", studentForWorkMap)
            .addAttribute("tempWork", new Student_work()));

        return "student_works";
    }

    @PostMapping("/works/save")
    public String save(@ModelAttribute("tempWork") Student_work
newStudentWork,
                        @ModelAttribute("newStudentList") List<Student>
newStudentList,
                        @NotNull HttpServletRequest request) {
        worksService.save(newStudentList, newStudentWork, (User)
request.getSession().getAttribute("ssouser"));
        return "redirect:/works";
    }
}

```

Приложение Ж. Код Представления для страницы «Работы»

```
<!DOCTYPE html>
<html lang="ru" xmlns="http://www.w3.org/1999/xhtml"
xmlns:th="http://www.thymeleaf.org">
<head>
  <meta charset="UTF-8">
  <title>Работы</title>
  <link th:href="@{/common.css}" rel="stylesheet" type="text/css">
  <style>

  </style>
</head>
<body>
<header th:insert="fragments/header :: header"></header>
<div id="works">
  <div th:each="work, iter : ${workList}">
    <p class="work-title" th:text="${iter.index + 1} + '. ' +
${work.title}"></p>
    <h5 th:text="${work.type.name}"></h5><br>
    <h4>Статус </h4>
    <h5 th:if="${work.completion_date}" th:text="'Окончена (' +
${work.completion_date.toLocaleString().substring(0,
work.completion_date.toLocaleString().length()-10)} + ')"></h5>
    <h5 th:if="${work.completion_date} == null">В разработке</h5><br>
    <div th:each="st : ${studentForWorkMap.get(work)}">
      <h5 th:text="${st.last_name} + ' ' + ${st.name.substring(0,1)} +
'.' + ${st.patronymic.substring(0,1)} + '.'"></h5>
    </div><br>
    <h4 th:with="t =
${scientificWorkList.get(__${iter.index}__).teacher}" th:text="'Научный
руководитель: ' + ${t.last_name} + ' ' + ${t.name.substring(0,1)} + '.' +
${t.patronymic.substring(0,1)} + '.'"></h4>
    <br>
    <br>
  </div>
</div>

<div id="create_work">
  <button id="create_work_button"></button>

  <form method="post" th:action="@{/works/save}">
    <h4>Тема</h4>
    <input type="text" th:field="*{tempWork.title}">
    <h4>Тип</h4>
    <input type="text" th:field="*{tempWork.type.name}">
    <h4>Студенты</h4>
    <input type="text" list="students">
    <datalist id="students">
      <option th:each="s : ${studentList}" th:value="${s.last_name} + '
' + ${s.name} + ' ' + ${s.patronymic}"></option>
    </datalist>

    <input type="submit" value="Создать">
  </form>
</div>

<details id="work_filters">
  <summary>Фильтровать</summary><br>

  <div id="teacher_filter">
    <h4>Научный руководитель </h4>
```

```

    <form method="post" id="form_teachers" onsubmit="return false">
      <label for="word_teachers"></label>
      <input type="text" id="word_teachers"
list="word_options_teachers">
      <div id="word_options_teachers">
        <div th:each="teacher : ${teacherList}" th:data-
value="${teacher.teacher_id}" th:data-text="${teacher.last_name} + ' ' +
${teacher.name} + ' ' + ${teacher.patronymic}">
          <p th:data-value="${teacher.email}"
th:text="${teacher.last_name} + ' ' + ${teacher.name} + ' ' +
${teacher.patronymic}"></p>
        </div>
      </div>
    </form>
  </div>

  <div id="student_filter">
    <h4>Студент </h4>

    <form method="post" id="form_students" onsubmit="return false">
      <label for="word_students"></label>
      <input type="text" id="word_students"
list="word_options_students">
      <div id="word_options_students">
        <div th:each="student : ${studentList}" th:data-
value="${student.getStudent_id()}">
          <p th:data-value="${student.email}"
th:text="${student.last_name} + ' ' + ${student.name} + ' ' +
${student.patronymic}"></p>
        </div>
      </div>
    </form>
  </div>

  <div id="title_filter">
    <h4>Тема </h4>
    <form method="post" id="form_titles" th:object="${tempWork}"
th:action="@{/works/filter/title}">
      <label for="word_titles"></label>
      <input type="text" id="word_titles" th:field="*{title}">
    </form>
  </div>
</details>
<script defer>
  // students
  let word_students = document.getElementById('word_students');
  let word_options_students =
document.getElementById('word_options_students');
  let studentList = word_options_students.querySelectorAll('div');
  studentList.forEach(student => {
    student.addEventListener('click', () => {
      console.log(student.dataset.value);
      window.location.href =
`/works/filter/student/${student.dataset.value}`;
    });
  });
  window.addEventListener('load', () => {

    word_students.addEventListener('input', (e) => {
      let el = e.target;
      if(el.value) {

```

```

        word_options_students.style.display = "block";
    } else word_options_students.style.display = "none";
    let t;
    let all = word_options_students.getElementsByTagName('div');
    let ifNoCatches = false;
    for (let i = 0; i < all.length; i++) {
        t = all[i];
        if(t.textContent.toUpperCase().includes(el.value.toUpperCase())) {
            t.style.display = "block";
            ifNoCatches = true;
        } else {
            t.style.display = "none";
        }
    }
    console.log(ifNoCatches);
    if(ifNoCatches === false) {
        word_options_students.value = "";
    }
    });
});
// teachers
let word_teachers = document.getElementById('word_teachers');
let word_options_teachers =
document.getElementById('word_options_teachers');
let teacherList = word_options_teachers.querySelectorAll('div');
teacherList.forEach(teacher => {
    teacher.addEventListener('click', () => {
        console.log(teacher.dataset.value);
        window.location.href =
`/works/filter/teacher/${teacher.dataset.value}`;
    });
});
window.addEventListener('load', () => {
    word_teachers.addEventListener('input', (e) => {
        let el = e.target;
        if(el.value) {
            word_options_teachers.style.display = "block";
        } else {
            word_options_teachers.style.display = "none";
        }
        let t;
        let all = word_options_teachers.getElementsByTagName('div');
        let ifNoCatches = false;
        for (let i = 0; i < all.length; i++) {
            t = all[i];
            if(t.textContent.toUpperCase().includes(el.value.toUpperCase())) {
                t.style.display = "block";
                ifNoCatches = true;
            } else {
                t.style.display = "none";
            }
        }
        console.log(ifNoCatches);
        if(ifNoCatches === false) {
            word_options_teachers.value = "";
        }
    });
});
});
</script>
</body>
</html>

```

Приложение И. Hibernate запросы при загрузке страницы «Работы»

1. Hibernate:

```
select teacher0.teacher_id as teacher_1_12_,
teacher0.acad_pos_id as acad_pos7_12_, teacher0.degree_id as
degree_i8_12_, teacher0.personal_id as personal6_12_ from teacher
teacher0
```
2. Hibernate:

```
select academic_p0.acad_pos_id as acad_pos1_0_0_,
academic_p0.name as name2_0_0_ from academic_position academic_p0_ where
academic_p0.acad_pos_id=?
```
3. Hibernate:

```
select science_de0.degree_id as degree_i1_5_0_,
science_de0.name as name2_5_0_ from science_degree science_de0_ where
science_de0.degree_id=?
```
4. Hibernate:

```
select department0.department_id as departme1_1_0_,
department0.name as name2_1_0_, department0.school_id as
school_i3_1_0_, school1.school_id as school_i1_4_1_, school1.name as
name2_4_1_ from department department0_ left outer join school school1_
on department0.school_id=school1.school_id where
department0.department_id=?
```
5. Hibernate:

```
select academic_p0.acad_pos_id as acad_pos1_0_0_,
academic_p0.name as name2_0_0_ from academic_position academic_p0_ where
academic_p0.acad_pos_id=?
```
6. Hibernate:

```
select science_de0.degree_id as degree_i1_5_0_,
science_de0.name as name2_5_0_ from science_degree science_de0_ where
science_de0.degree_id=?
```
7. Hibernate:

```
select department0.department_id as departme1_1_0_,
department0.name as name2_1_0_, department0.school_id as
school_i3_1_0_, school1.school_id as school_i1_4_1_, school1.name as
name2_4_1_ from department department0_ left outer join school school1_
on department0.school_id=school1.school_id where
department0.department_id=?
```
8. Hibernate:

```
select academic_p0.acad_pos_id as acad_pos1_0_0_,
academic_p0.name as name2_0_0_ from academic_position academic_p0_ where
academic_p0.acad_pos_id=?
```
9. Hibernate:

```
select science_de0.degree_id as degree_i1_5_0_,
science_de0.name as name2_5_0_ from science_degree science_de0_ where
science_de0.degree_id=?
```
10. Hibernate:

```
select department0.department_id as departme1_1_0_,
department0.name as name2_1_0_, department0.school_id as
school_i3_1_0_, school1.school_id as school_i1_4_1_, school1.name as
name2_4_1_ from department department0_ left outer join school school1_
on department0.school_id=school1.school_id where
department0.department_id=?
```

11. Hibernate: select academic_p0.acad_pos_id as acad_pos1_0_0_,
academic_p0.name as name2_0_0_ from academic_position academic_p0_ where
academic_p0.acad_pos_id=?
12. Hibernate: select science_de0.degree_id as degree_i1_5_0_,
science_de0.name as name2_5_0_ from science_degree science_de0_ where
science_de0.degree_id=?
13. Hibernate: select department0.department_id as departme1_1_0_,
department0.name as name2_1_0_, department0.school_id as
school_i3_1_0_, school1.school_id as school_i1_4_1_, school1.name as
name2_4_1_ from department department0_ left outer join school school1_
on department0.school_id=school1.school_id where
department0.department_id=?
14. Hibernate: select academic_p0.acad_pos_id as acad_pos1_0_0_,
academic_p0.name as name2_0_0_ from academic_position academic_p0_ where
academic_p0.acad_pos_id=?
15. Hibernate: select science_de0.degree_id as degree_i1_5_0_,
science_de0.name as name2_5_0_ from science_degree science_de0_ where
science_de0.degree_id=?
16. Hibernate: select department0.department_id as departme1_1_0_,
department0.name as name2_1_0_, department0.school_id as
school_i3_1_0_, school1.school_id as school_i1_4_1_, school1.name as
name2_4_1_ from department department0_ left outer join school school1_
on department0.school_id=school1.school_id where
department0.department_id=?
17. Hibernate: select student0.student_id as student_1_9_,
student0.email as email2_9_, student0.group_id as group_id7_9_,
student0.last_name as last_nam3_9_, student0.name as name4_9_,
student0.patronymic as patronym5_9_, student0.personal_id as
personal6_9_ from student student0_
18. Hibernate: select group0.group_id as group_id1_2_0_,
group0.course as course2_2_0_, group0.enrollment_year as
enrollme3_2_0_, group0.name as name4_2_0_, group0.speciality_id as
speciali5_2_0_, speciality1.speciality_id as speciali1_8_1_,
speciality1.department_id as departme3_8_1_, speciality1.name as
name2_8_1_, department2.department_id as departme1_1_2_,
department2.name as name2_1_2_, department2.school_id as school_i3_1_2_
from group_ group0_ left outer join speciality speciality1_ on
group0.speciality_id=speciality1.speciality_id left outer join
department department2_ on
speciality1.department_id=department2.department_id where
group0.group_id=?

19. Hibernate: select group0_.group_id as group_id1_2_0_, group0_.course as course2_2_0_, group0_.enrollment_year as enrollme3_2_0_, group0_.name as name4_2_0_, group0_.speciality_id as speciali5_2_0_, speciality1_.speciality_id as speciali1_8_1_, speciality1_.department_id as departme3_8_1_, speciality1_.name as name2_8_1_, department2_.department_id as departme1_1_2_, department2_.name as name2_1_2_, department2_.school_id as school_i3_1_2_ from group_ group0_ left outer join speciality speciality1_ on group0_.speciality_id=speciality1_.speciality_id left outer join department department2_ on speciality1_.department_id=department2_.department_id where group0_.group_id=?
20. Hibernate: select group0_.group_id as group_id1_2_0_, group0_.course as course2_2_0_, group0_.enrollment_year as enrollme3_2_0_, group0_.name as name4_2_0_, group0_.speciality_id as speciali5_2_0_, speciality1_.speciality_id as speciali1_8_1_, speciality1_.department_id as departme3_8_1_, speciality1_.name as name2_8_1_, department2_.department_id as departme1_1_2_, department2_.name as name2_1_2_, department2_.school_id as school_i3_1_2_ from group_ group0_ left outer join speciality speciality1_ on group0_.speciality_id=speciality1_.speciality_id left outer join department department2_ on speciality1_.department_id=department2_.department_id where group0_.group_id=?
21. Hibernate: select student_wo0_.student_id as student_1_10_, student_wo0_.work_id as work_id2_10_, student_wo0_.completion_date as completi3_10_, student_wo0_.leave_reason as leave_re4_10_, student_wo0_.score as score5_10_, student_wo0_.start_date as start_da6_10_, student_wo0_.title as title7_10_, student_wo0_.type as type8_10_ from student_work student_wo0_
22. Hibernate: select student_wo0_.type as typel_11_0_, student_wo0_.name as name2_11_0_ from student_work_type student_wo0_ where student_wo0_.type=?
23. Hibernate: select student_wo0_.type as typel_11_0_, student_wo0_.name as name2_11_0_ from student_work_type student_wo0_ where student_wo0_.type=?
24. Hibernate: select scientific0_.work_id as work_id1_7_, scientific0_.teacher_id as teacher_3_7_, scientific0_.title as title2_7_ from scientific_work scientific0_
25. Hibernate: select student_wo0_.student_id as student_1_10_, student_wo0_.work_id as work_id2_10_, student_wo0_.completion_date as

- completi3_10_, student_wo0_.leave_reason as leave_re4_10_, student_wo0_.score as score5_10_, student_wo0_.start_date as start_da6_10_, student_wo0_.title as title7_10_, student_wo0_.type as type8_10_ from student_work student_wo0_
26. Hibernate: select scientific0_.work_id as work_id1_7_, scientific0_.teacher_id as teacher_3_7_, scientific0_.title as title2_7_ from scientific_work scientific0_
27. Hibernate: select student_wo0_.student_id as student_1_10_, student_wo0_.work_id as work_id2_10_, student_wo0_.completion_date as completi3_10_, student_wo0_.leave_reason as leave_re4_10_, student_wo0_.score as score5_10_, student_wo0_.start_date as start_da6_10_, student_wo0_.title as title7_10_, student_wo0_.type as type8_10_ from student_work student_wo0_ where student_wo0_.work_id=?
28. Hibernate: select student0_.student_id as student_1_9_, student0_.email as email2_9_, student0_.group_id as group_id7_9_, student0_.last_name as last_nam3_9_, student0_.name as name4_9_, student0_.patronymic as patronym5_9_, student0_.personal_id as personal6_9_ from student student0_ where student0_.student_id=?
29. Hibernate: select student_wo0_.student_id as student_1_10_, student_wo0_.work_id as work_id2_10_, student_wo0_.completion_date as completi3_10_, student_wo0_.leave_reason as leave_re4_10_, student_wo0_.score as score5_10_, student_wo0_.start_date as start_da6_10_, student_wo0_.title as title7_10_, student_wo0_.type as type8_10_ from student_work student_wo0_ where student_wo0_.work_id=?
30. Hibernate: select student0_.student_id as student_1_9_, student0_.email as email2_9_, student0_.group_id as group_id7_9_, student0_.last_name as last_nam3_9_, student0_.name as name4_9_, student0_.patronymic as patronym5_9_, student0_.personal_id as personal6_9_ from student student0_ where student0_.student_id=?
31. Hibernate: select student_wo0_.student_id as student_1_10_, student_wo0_.work_id as work_id2_10_, student_wo0_.completion_date as completi3_10_, student_wo0_.leave_reason as leave_re4_10_, student_wo0_.score as score5_10_, student_wo0_.start_date as start_da6_10_, student_wo0_.title as title7_10_, student_wo0_.type as type8_10_ from student_work student_wo0_ where student_wo0_.work_id=?
32. Hibernate: select student0_.student_id as student_1_9_, student0_.email as email2_9_, student0_.group_id as group_id7_9_, student0_.last_name as last_nam3_9_, student0_.name as name4_9_, student0_.patronymic as patronym5_9_, student0_.personal_id as personal6_9_ from student student0_ where student0_.student_id=?

33. Hibernate: select student_wo0_.student_id as student_1_10_, student_wo0_.work_id as work_id2_10_, student_wo0_.completion_date as completi3_10_, student_wo0_.leave_reason as leave_re4_10_, student_wo0_.score as score5_10_, student_wo0_.start_date as start_da6_10_, student_wo0_.title as title7_10_, student_wo0_.type as type8_10_ from student_work student_wo0_ where student_wo0_.work_id=?
34. Hibernate: select student0_.student_id as student_1_9_, student0_.email as email2_9_, student0_.group_id as group_id7_9_, student0_.last_name as last_nam3_9_, student0_.name as name4_9_, student0_.patronymic as patronym5_9_, student0_.personal_id as personal6_9_ from student student0_ where student0_.student_id=?
35. Hibernate: select student0_.student_id as student_1_9_, student0_.email as email2_9_, student0_.group_id as group_id7_9_, student0_.last_name as last_nam3_9_, student0_.name as name4_9_, student0_.patronymic as patronym5_9_, student0_.personal_id as personal6_9_ from student student0_ where student0_.student_id=?
36. Hibernate: select student_wo0_.student_id as student_1_10_, student_wo0_.work_id as work_id2_10_, student_wo0_.completion_date as completi3_10_, student_wo0_.leave_reason as leave_re4_10_, student_wo0_.score as score5_10_, from student_work student_wo0_ where student_wo0_.work_id=?
37. Hibernate: select student_wo0_.student_id as student_1_10_, student_wo0_.work_id as work_id2_10_, student_wo0_.completion_date as completi3_10_, student_wo0_.leave_reason as leave_re4_10_, student_wo0_.score as score5_10_, from student_work student_wo0_
38. Hibernate: select scientific0_.work_id as work_id1_7_, scientific0_.teacher_id as teacher_3_7_, scientific0_.title as title2_7_ from scientific_work scientific0_
39. Hibernate: select student_wo0_.student_id as student_1_10_, student_wo0_.work_id as work_id2_10_, student_wo0_.completion_date as completi3_10_, student_wo0_.leave_reason as leave_re4_10_, student_wo0_.score as score5_10_, from student_work student_wo0_
40. Hibernate: select scientific0_.work_id as work_id1_7_, scientific0_.teacher_id as teacher_3_7_, scientific0_.title as title2_7_ from scientific_work scientific0_

Приложение К. Ссылка на репозиторий Github с исходным кодом

