

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
«МОСКОВСКИЙ ПОЛИТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»
(МОСКОВСКИЙ ПОЛИТЕХ)

ДОПУЩЕНА К ЗАЩИТЕ

Заведующий кафедрой

_____ / Пухова Е. А. /

Руководитель образовательной программы

_____ / Даньшина М. В. /

ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА

по теме:

ВЕБ-ПРИЛОЖЕНИЕ ДЛЯ АВТОМАТИЗАЦИИ УПРАВЛЕНИЯ ВЫ-
ПУСКНЫМИ КВАЛИФИКАЦИОННЫМИ РАБОТАМИ

по направлению 09.03.03 Прикладная информатика

Образовательная программа (профиль)

«Информационные технологии управления бизнесом»

Студент: _____ / Казарян Ашот Сергеевич, 221-365 /
подпись *ФИО, группа*

Руководитель ВКР: _____ / Осьмин Владимир Вячеславович /
подпись *ФИО, уч. звание и степень*

Москва 2026

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
«МОСКОВСКИЙ ПОЛИТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»
(МОСКОВСКИЙ ПОЛИТЕХ)

ЗАДАНИЕ НА ВЫПУСКНУЮ КВАЛИФИКАЦИОННУЮ РАБОТУ
по направлению 09.03.03 Прикладная информатика
Образовательная программа (профиль)
«Информационные технологии управления бизнесом»

Тема ВКР	Веб-приложение для автоматизации управления выпускными квалификационными работами
ПРАКТИЧЕСКИЙ РЕЗУЛЬТАТ	
Назначение	Веб-приложение для автоматизации распределения студентов выпускных курсов по научным руководителям и утверждения тем выпускных квалификационных работ. Студентам система предоставляет информацию о вакантных местах на руководство ВКР у руководителей и позволяет подавать заявки. Руководители управляют входящими заявками, отслеживают заполненность своей квоты и при необходимости запрашивают её изменение. Заведующие кафедрами осуществляют контроль процесса, формируют реестры и проекты приказов. Администраторы управляют справочниками системы.
Основные функции	1. Авторизация пользователей с разграничением ролей (студент, руководитель, заведующий кафедрой, администратор). 2. Просмотр студентом списка руководителей своей кафедры с индикацией количества вакантных мест на руководство ВКР. 3. Подача студентом заявки на руководство ВКР с указанием выбранного руководителя, темы, описания работы и используемых технологий. 4. Автоматическая проверка наличия вакантных мест у руководителя перед созданием заявки. 5. Просмотр руководителем входящих заявок с возможностью одобрения или отклонения.

	6. Автоматическое обновление счётчика закреплённых студентов при одобрении заявки и его уменьшение при отзыве одобренной заявки. 7. Отзыв заявки студентом или руководителем на любом этапе. 8. Восстановление отозванной заявки студентом или руководителем. 9. Автоматическое ведение журнала изменений для каждой заявки. 10. Запрос руководителем изменения квоты на руководство ВКР с обоснованием; рассмотрение и одобрение или отклонение запроса заведующим кафедры. 11. Экспорт реестра заявок в формате Excel. 12. Генерация проекта приказа об утверждении тем и назначении руководителей в формате Word по шаблону университета с группировкой по учебным группам. 13. Административная панель для управления справочниками. 14. Роль заведующего кафедры с доступом только к данным своей кафедры и правом обработки запросов на изменение квоты у руководителя и формирования проектов приказов.
Используемые технологии и платформы	База данных: SQLite Серверная часть: Python, Django Клиентская часть: HTML, CSS, JavaScript, Bootstrap

ВЫПОЛНЕНИЕ РАБОТЫ	
Решаемые задачи	1. Провести анализ предметной области и существующего процесса управления ВКР. 2. Разработать ролевую модель и личные кабинеты пользователей. 3. Реализовать модуль управления заявками на руководство ВКР. 4. Разработать модули автоматического учёта вакантных мест и управления квотой научного руководителя. 5. Разработать модуль экспорта данных и генерации проектов приказов. 6. Реализовать административную панель с разграничением прав доступа для ведения справочников и мониторинга данных по кафедре. 7. Провести функциональное тестирование системы и оценку эффективности разработанной системы.
Состав технической документации	1. Пояснительная записка. 2. Техническое задание.
Состав графической части	1. Диаграмма прецедентов (UML): 1 экз. 2. Диаграмма последовательностей (UML): 1 экз. 3. ER-диаграмма (инфологическая модель): 1 экз. 4. ER-диаграмма (даталогическая модель): 1 экз. 5. BPMN-диаграмма (AS-IS): 1 экз. 6. BPMN-диаграмма (TO-BE): 1 экз. 7. IDEF0-диаграмма (TO-BE, контекстная A-0): 1 экз. 8. IDEF0-диаграмма (TO-BE, декомпозиция A0): 1 экз. 9. Диаграмма процесса (TO-BE, EPC): 1 экз. 10. Скриншоты пользовательского интерфейса и сформированных документов: 12 экз.

ПЛАН РАБОТЫ НАД ВКР

Этапы	Недели семестра																	
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
Анализ предметной области и существующего процесса																		
Обзор и сравнительный анализ аналогов																		
Формулирование требований к системе																		
Проектирование бизнес-процессов																		
Проектирование архитектуры и выбор технологий																		
Проектирование структуры базы данных																		
Реализация модулей аутентификации и прав доступа																		
Реализация управления заявками и учёта количества закреплённых за руководителем студентов																		
Реализация механизма запроса изменения квоты																		
Реализация отзыва и восстановления заявок																		
Реализация экспорта реестров и генерации проектов приказов																		
Тестирование и оценка эффективности																		
Оформление пояснительной записки																		

РУКОВОДИТЕЛЬ ОП:

«__»____2026, _____ / Даньшина Марина Владимировна /
подпись *ФИО, уч. звание и степень*

РУКОВОДИТЕЛЬ ВКР:

«__»____2026, _____ / Осьмин Владимир Вячеславович /
подпись *ФИО, уч. звание и степень*

СТУДЕНТ:

«__»____2026, _____ / Казарян Ашот Сергеевич, 221–365/
подпись *ФИО, группа*

АННОТАЦИЯ

Выпускная квалификационная работа посвящена разработке веб-приложения для автоматизации управления выпускными квалификационными работами в высшем учебном заведении.

Цель настоящей работы разработать веб-приложение для автоматизации управления выпускными квалификационными работами.

Объект исследования — процессы организации и управления выпускными квалификационными работами в университете. Предмет исследования — методы и средства автоматизации управления ВКР, включая веб-технологии, базы данных и генерацию документов.

Работа содержит три главы, Введение, Заключение и включает 91 страницу, 25 рисунков, 11 таблиц, 59 источников.

Во Введении обоснована актуальность, сформулированы цель и задачи, определены объект и предмет исследования, показаны практическая значимость и новизна.

В первой главе приведено описание заказчика, описаны существующие бизнес-процессы (AS-IS), проведён обзор программных решений, сформулированы требования к системе и выявлены недостатки действующего процесса.

Во второй главе обоснован выбор архитектуры и технологического стека, выполнено проектирование информационной системы, разработаны модели данных и бизнес-процессов, описаны сценарии использования и бизнес-логика, выполнено проектирование пользовательского интерфейса.

В третьей главе представлены методика и результаты локального тестирования, сценарии проверки критических путей и направления для дальнейшего развития.

В Заключении подведены итоги работы, приведены результаты оценки эффективности и определены направления дальнейшего развития системы.

СОДЕРЖАНИЕ

ВВЕДЕНИЕ	8
1 АНАЛИЗ ПРЕДМЕТНОЙ ОБЛАСТИ И БИЗНЕС-ТРЕБОВАНИЙ	13
1.1 Описание заказчика и анализ существующих бизнес-процессов	13
1.2 Обзор и сравнительный анализ существующих программных решений..	15
1.3 Формирование целей и задач, формализация требований.....	18
1.4 Анализ проблем и недостатков действующей системы.....	20
1.5 Выводы	21
2 ПРОЕКТИРОВАНИЕ И БИЗНЕС-ЛОГИКА ПРИЛОЖЕНИЯ	23
2.1 Выбор архитектурного подхода и технологий разработки	23
2.2 Формализация требований и проектирование модели данных	24
2.3 Описание бизнес-процессов и бизнес-логики.....	29
2.4 Проектирование пользовательского интерфейса.....	35
2.5 Выводы	37
3 ТЕХНИЧЕСКАЯ РЕАЛИЗАЦИЯ И ОЦЕНКА ЭФФЕКТИВНОСТИ	38
3.1 Реализация серверной части	38
3.2 Разработка клиентской части.....	42
3.3 Тестирование и направления для развития	43
3.4 Оценка эффективности разработанной системы	45
3.5 Выводы	46
ЗАКЛЮЧЕНИЕ	48
СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ	51
ПРИЛОЖЕНИЕ А	58
ПРИЛОЖЕНИЕ Б.....	77
ПРИЛОЖЕНИЕ В	86
ПРИЛОЖЕНИЕ Г.....	87
ПРИЛОЖЕНИЕ Д.....	88

ВВЕДЕНИЕ

Выпускная квалификационная работа (далее – ВКР) представляет собой самостоятельное комплексное исследование, демонстрирующее уровень подготовки выпускника, его способность решать профессиональные задачи и осуществлять выбранный вид деятельности [1].

Тема настоящей работы – «Веб-приложение для автоматизации управления выпускными квалификационными работами». Выбор темы обусловлен необходимостью совершенствования и автоматизации организационных процессов, связанных с процессом управления ВКР в высших учебных заведениях. Применяемый в настоящее время подход к управлению ВКР является трудоёмким, характеризуется недостаточной информированностью студентов выпускных курсов о количестве вакантных мест у научных руководителей и подвержен ошибкам при ручном формировании документов. Студенты выпускных курсов выбирают научного руководителя, не имея актуальной информации о количестве вакантных у него мест для закрепления студентов. Данные о заполненности мест ведутся сотрудниками кафедры в разрозненных файловых хранилищах и не всегда доступны студентам в структурированном виде. Это приводит к тому, что список студентов, закреплённых за каждым руководителем, формируется неравномерно и не всегда своевременно. Руководители не имеют централизованного инструмента для обработки входящих обращений, а сотрудникам кафедр приходится вручную формировать сводные таблицы и проекты приказов [2–5]. Такой подход, обозначаемый как процесс «как есть» (AS-IS), является трудоёмким и подверженным ошибкам, что снижает общую эффективность организационного процесса.

Согласно данным Правительства РФ, в 2024 году российские университеты окончили около 824 тысяч студентов по программам бакалавриата, специалитета и магистратуры [6, 7]. Это означает, что ежегодно сотни тысяч выпускников проходят через процедуру выбора тем и научных руководителей, а кафедры обрабатывают сотни обращений и готовят десятки приказов. В таких

условиях отсутствие специализированных средств автоматизации приводит к значительным временным и трудовым затратам, что подтверждает актуальность разработки веб-приложения для автоматизации управления ВКР.

Анализ существующих программных решений показал, что специализированные средства автоматизации управления ВКР представлены ограниченно, а используемые на практике системы либо не обеспечивают необходимую функциональность, либо требуют существенной адаптации под особенности конкретного учебного заведения. Существующие платформы либо не обладают необходимой функциональностью, либо требуют трудоёмкой и дорогостоящей доработки. Наиболее распространённым способом организации остаются электронные таблицы, которые не являются специализированным решением для управления ВКР и не автоматизируют ключевые бизнес-процессы. Таким образом, актуальность работы заключается в создании единой специализированной веб-системы, устраняющей перечисленные недостатки и обеспечивающей автоматизацию цикла управления ВКР.

Объектом исследования является процесс управления ВКР в высшем учебном заведении. Предмет исследования – методы и средства автоматизации управления ВКР, включая веб-технологии, базы данных и автоматизированное формирование документов.

Работа включает анализ существующего процесса (AS-IS) и разработку целевой системы (TO-BE), а также тестирование программного продукта.

Целью настоящей работы является разработка веб-приложения для автоматизации управления выпускными квалификационными работами. Для этого разрабатывается программный продукт, автоматизирующий ключевые процессы управления ВКР.

Для достижения поставленной цели необходимо решить следующие задачи:

1. Провести анализ предметной области и существующего процесса управления ВКР.
2. Разработать ролевую модель и личные кабинеты пользователей.

3. Реализовать модуль управления заявками на руководство ВКР.
4. Разработать модули автоматического учёта вакантных мест и управления квотой научного руководителя.
5. Разработать модуль экспорта данных и генерации проектов приказов.
6. Реализовать административную панель с разграничением прав доступа для ведения справочников и мониторинга данных по кафедре.
7. Провести функциональное тестирование системы и оценку эффективности разработанной системы.

Практическим результатом работы является веб-приложение, которое функционирует на базе программной платформы Django и системе управления базами данных (далее – СУБД) SQLite и обеспечивает:

1. Личный кабинет студента с возможностью выбора руководителя, просмотра количества вакантных у него мест на руководство ВКР и подачи заявки с указанием темы, описания и используемых технологий.
2. Личный кабинет руководителя с отображением входящих заявок, возможностью их одобрения или отклонения, а также функцией запроса изменения максимально возможного количества закреплённых студентов.
3. Автоматический учёт вакантных мест руководителя с предотвращением превышения максимально возможного количества закреплённых студентов и корректным пересчётом при отзыве заявок.
4. Механизм двустороннего отзыва заявки студентом или руководителем на любом этапе с возможностью последующего восстановления.
5. Административный интерфейс для управления пользователями и заявками с разграничением прав доступа, включая отдельную роль заведующего кафедрой.
6. Ведение журнала истории для каждой заявки.
7. Экспорт реестра заявок в формате Excel
8. Формирование проекта приказа об утверждении тем и назначении руководителей в формате Word по шаблону университета.

Практическая значимость работы заключается в возможности использования разработанного приложения на кафедре или факультете для автоматизации процесса управления ВКР, что позволит сократить временные затраты сотрудников, повысить информированность участников о ходе распределения студентов по научным руководителям и достоверность информации, а также снизить риск ошибок при формировании документов. Элементами новизны разработанного решения являются следующие механизмы, реализованные в рамках веб-ориентированной модели управления жизненным циклом заявки на руководство ВКР:

- 1) автоматический учёт вакантных мест научных руководителей с контролем максимально возможного количества закреплённых студентов и предотвращением их превышения;
- 2) двусторонний отзыв и восстановление заявок на любом этапе их жизненного цикла с корректным пересчётом нагрузки;
- 3) генерация проектов приказов в формате Word по утверждённому шаблону университета с группировкой по учебным группам.

В рамках настоящей работы используются следующие основные понятия:

- 1) руководитель – работник университета, за которым приказом закрепляется студент для подготовки выпускной квалификационной работы; научный руководитель;
- 2) студент – лицо, осваивающее образовательную программу высшего образования и выполняющее выпускную квалификационную работу; студент выпускного курса;
- 3) квота – максимальное количество студентов, которых руководитель может принять на руководство ВКР; максимально возможно количество вакантных мест руководителя на руководство ВКР;
- 4) двусторонний отзыв – возможность отзыва заявки студентом или руководителем на любом этапе её жизненного цикла. Механизм восстановления

позволяет восстановить отозванную заявку в работу той стороной, которая её отозвала.

Дополнительные материалы по разработанному веб-приложению размещены в публичном репозитории и доступны для ознакомления по следующему адресу: <https://github.com/AshotKazarian/diplom2026>.

В соответствии с требованиями образовательной программы «Информационные технологии управления бизнесом» [8] практическим результатом ВКР является информационная система, обладающая актуальностью, завершенностью, практической ценностью, новизной, сложностью, технологичностью и экономичностью. Работа выполнена в полном соответствии с требованиями [9], предъявляемыми к выпускным квалификационным работам по направлению 09.03.03 «Прикладная информатика», и содержит все необходимые компоненты: практический результат, пояснительную записку и презентационные материалы.

1 АНАЛИЗ ПРЕДМЕТНОЙ ОБЛАСТИ И БИЗНЕС-ТРЕБОВАНИЙ

1.1 Описание заказчика и анализ существующих бизнес-процессов

Заказчиком разрабатываемой системы выступает кафедра «Инфокогнитивные технологии» федерального государственного автономного образовательного учреждения высшего образования «Московский политехнический университет» (далее – Московский политехнический университет). В рамках рассматриваемой предметной области кафедра осуществляет организацию и контроль процесса подготовки выпускных квалификационных работ студентов выпускных курсов, включая распределение студентов по научным руководителям, контроль квот научных руководителей и подготовку приказов о назначении руководителей и утверждении тем выпускных квалификационных работ.

Организационная структура управления процессом ВКР включает следующие роли:

1) заведующий кафедрой осуществляет общее руководство, утверждение приказов, а также рассмотрение и согласование запросов руководителей на изменение их квот; координирует распределение студентов по руководителям;

2) научные руководители рассматривают обращения студентов и при необходимости инициируют предложения по изменению квот;

3) студенты выпускных курсов направляют обращения на руководство ВКР, предлагают тему и узнают о решении руководителя.

По результатам опроса 5 представителей кафедры и наблюдения за процессом распределения студентов по научным руководителям установлено, что существующий процесс характеризуется высокой трудоёмкостью и недостаточной информированностью его участников.

Основные этапы текущего процесса распределения студентов по научным руководителям представлены на рисунке 1.



Рисунок 1 – Основные этапы текущего процесса распределения студентов по руководителям ВКР

Анализ представленного процесса показывает необходимость учёта потребностей всех его участников. Целевая аудитория разрабатываемой системы и её основные потребности представлены в таблице 1.

На рисунке А.1 представлена диаграмма текущего бизнес-процесса «Управление заявками на руководство ВКР» в нотации BPMN 2.0 (модель AS-IS). Диаграмма отражает существующий порядок взаимодействия между студентом, научным руководителем и заведующим кафедрой.

Таблица 1 – Характеристика целевой аудитории системы

Группа пользователей	Основные потребности
1	2
Студенты	Механизм выбора научного руководителя с учётом количества вакантных у него мест, подача заявки, отслеживание статуса, отзыв и восстановление заявки
Руководители	Управление входящими заявками, контроль количества вакантных мест на руководство ВКР, запрос изменения квоты

Продолжение таблицы 1

1	2
Заведующие кафедрами	Контроль процесса в рамках своей кафедры, обработка запросов на изменение квоты, формирование проектов приказов
Администратор	Управление справочниками системы

Ключевые особенности текущего процесса:

- 1) обращения подаются через личное обращение или средства связи;
- 2) данные о студентах и руководителях ведутся в электронных таблицах вручную;
- 3) отсутствует автоматический контроль вакантных мест у руководителей;
- 4) отсутствие единого источника актуальной информации, доступного всем участникам процесса.

Таким образом, существующий процесс является трудоёмким, характеризуется низкой оперативностью и недостаточной информированностью участников о текущем распределении студентов выпускных курсов по научным руководителям. Указанные недостатки приводят к увеличению временных затрат сотрудников кафедры и снижению удовлетворённости участников процесса. Это обосновывает целесообразность разработки автоматизированной системы.

Разрабатываемая система направлена на устранение выявленных недостатков за счёт единого веб-пространства, где студенты будут оперативно видеть количество вакантных мест и область научных интересов руководителей и подавать заявки, руководители – управлять входящими заявками, а заведующие кафедрами – автоматически формировать реестры и приказы, минимизируя ручной труд и ошибки.

1.2 Обзор и сравнительный анализ существующих программных решений

Специализированные программные продукты для управления процессом распределения студентов по научным руководителям ВКР отсутствуют,

поэтому для сравнения были выбраны наиболее распространённые инструменты, используемые образовательными организациями для решения данной задачи косвенным способом.

Для сравнительного анализа были выбраны три типа решений, наиболее часто используемых в практике российских университетов, которые представлены ниже в порядке убывания их распространённости и практической значимости для рассматриваемой предметной области:

- 1) решения на базе электронных таблиц;
- 2) LMS Moodle;
- 3) 1С:Университет.

Критерии сравнения и результаты анализа представлены в таблице 2.

Таблица 2 – Сравнительный анализ аналогов

Критерий	LMS Moodle	1С:Университет	Электронные таблицы	Разрабатываемая система
Управление заявками студентов	Требуется доработки	Ограничено	Ручной ввод	Полная автоматизация
Учет вакантных мест у руководителей	Нет	Частично	Нет	Полная автоматизация
Запрос на изменение квоты от руководителя	Нет	Нет	Нет	Да
Генерация документов (приказы, реестры)	Нет	Да	Да	Да
Информированность участников о статусе распределения студентов по руководителям	Нет	Частично	Нет	Да
Возможность адаптации под конкретный университет	Требуется программирования	Требуется программирования	Да	Гибкая настройка
Стоимость внедрения	Бесплатно	Высокая	Бесплатно	Временные затраты на разработку

Обобщённый анализ функциональных возможностей рассмотренных решений, а также опыт внедрения информационных систем в российских университетах показывают, что существующие решения не обеспечивают комплексной автоматизации рассматриваемого процесса [10–12]. Данные таблицы 2 усиливают этот вывод.

Их ключевыми недостатками являются:

- 1) отсутствие единой цифровой среды и централизованного хранения данных;
- 2) ручной документооборот и высокий риск ошибок;
- 3) низкая информированность участников о состоянии процесса распределения студентов по руководителям;
- 4) отсутствие автоматического учёта вакантных мест у руководителей;
- 5) трудоёмкость адаптации к специфике конкретного учебного заведения;
- 6) высокая стоимость внедрения или необходимость существенной доработки.

Как показывают данные таблицы 2, наиболее распространённый на практике подход, а именно использование электронных таблиц, не является специализированным решением и не обеспечивает автоматизации ключевых бизнес-процессов управления ВКР.

Для более полной оценки целесообразности разработки такого решения был проведён SWOT-анализ, позволяющий системно оценить сильные и слабые стороны разработки, а также внешние возможности и угрозы. Результаты анализа представлены на рисунке 2.

Проведённый SWOT-анализ показывает, что сильные стороны проекта и внешние возможности преобладают над слабыми сторонами и угрозами, что подтверждает обоснованность разработки специализированного веб-приложения.

Таким образом, на основе проведённого анализа принимается решение о разработке веб-приложения, ориентированного на специфику управления ВКР в Московском политехническом университете. Данное решение обусловлено отсутствием на рынке специализированных аналогов, полностью покрывающих выявленные потребности кафедры.

Разрабатываемое веб-приложение обеспечит автоматизацию подачи и обработки заявок, учёт вакантных мест руководителей, а также генерацию документов по шаблонам университета. Важным преимуществом станет отсутствие необходимости в приобретении дополнительных лицензий благодаря использованию открытого стека технологий.

Сильные стороны	Слабые стороны
1. Покрытие специфических потребностей в управлении ВКР. 2. Отсутствие затрат на приобретение лицензий ввиду применения открытого стека технологий. 3. Гибкая настройка справочников без необходимости изменения программного кода. 4. Механизмы, отсутствующие в аналогах. 5. Журнал изменений, обеспечивающий аудит всех операций.	1. Отсутствие интеграции с информационными системами университета на текущем этапе разработки. 2. Ограниченный функционал документооборота.
Возможности	Угрозы
1. Масштабирование на другие кафедры и факультеты университета. 2. Интеграция с архитектурой университета. 3. Расширение перечня формируемых документов. 4. Применение системы в других университетах.	1. Изменение внутренних регламентов университета. 2. Появление конкурирующих программных продуктов. 3. Сопротивление сотрудников переходу на новую систему.

Рисунок 2 – SWOT-анализ разрабатываемой системы

1.3 Формирование целей и задач, формализация требований

Целью разрабатываемой системы является автоматизация ключевых этапов управления ВКР: от подачи заявки студентом до формирования проекта приказа заведующим кафедрой.

Для оценки ожидаемого эффекта от внедрения системы определены следующие направления улучшения процесса:

1. Сокращение временных затрат на обработку заявок и формирование документов.
2. Обеспечение информированности всех участников о статусе заявок и вакантных мест у руководителей.

3. Исключение ошибок, связанных с человеческим фактором.

4. Автоматический учёт количества вакантных мест у руководителей.

Для оценки эффективности разработанной системы определены следующие целевые показатели, характеризующие переход от текущего состояния (AS-IS) к проектному (TO-BE). Данные показатели сформулированы по методике SMART [13–15] и представлены в таблице 3.

Базовые значения показателей определены на основе наблюдения за процессом распределения студентов по руководителям ВКР и экспертной оценки 5 сотрудников кафедры. Целевые значения получены расчётным путём исходя из предполагаемого исключения ручных операций после внедрения автоматизированной системы. Достоверность экспертных оценок подтверждается многолетним опытом работы опрошенных сотрудников в процессе распределения студентов по руководителям. Расчёт целевых значений базируется на полном исключении ручных операций при автоматизации подачи заявок, учёта квот и генерации документов.

Недостаток НЗ (хранение данных в разрозненных электронных таблицах и формах) не привязан к отдельной строке таблицы, поскольку носит системный характер и устраняется созданием единого информационного пространства.

На основе выявленных недостатков действующего процесса и сформулированных требований к новой системе определены целевые показатели эффективности: сокращение времени выбора и закрепления научного руководителя с 5 до 1 суток, снижение трудоёмкости формирования проекта приказа с 4 до 0,2 рабочих часов, уменьшение доли ошибок при вводе данных с 10–15% до 1%. Достижение указанных показателей обеспечивается автоматизацией подачи заявок, централизованным учётом квот и генерацией выходных документов по утверждённым шаблонам.

Формирование требований стало ключевым этапом перехода от анализа недостатков действующего процесса к проектированию будущей системы.

Таблица 3 – Целевые показатели эффективности (ТО-ВЕ)

Тактическая цель	Измеримая задача (KPI)	Единица измерения	Базовое значение (AS-IS)	Целевое значение (TO-BE)	Привязка к проблеме (см. таблицу 5 в параграфе 1.4)
Сократить время выбора и закрепления научного руководителя	Среднее время от начала выбора руководителя студентом до подтверждения закрепления приказом	сутки	До 5	До 1	Н1
Снизить трудовые затраты на формирование приказа	Сокращение времени на формирование одного приказа	рабочие часы	4	До 0,2	Н4
Исключить ошибки ручного ввода	Снижение доли ошибок в данных	%	10-15	До 1	Н5
Обеспечить оперативное получение информации	Время получения сводки о статусе заявок и занятости руководителей	минуты	300	До 1	Н2

Требования к разрабатываемой системе были сформированы на основе анализа существующего процесса (AS-IS) и потребностей целевой аудитории. Выявленные требования классифицированы по уровням и типам в таблице Б.1.

1.4 Анализ проблем и недостатков действующей системы

Сопоставление текущего процесса управления ВКР со сформулированными в параграфе 1.3 требованиями позволяет выявить ряд существенных недостатков, которые классифицированы по характеру их возникновения и представлены в таблице 4.

Помимо перечисленных в таблице недостатков существующий процесс не предусматривает возможности корректировки уже совершённых действий: любая ошибка при заполнении данных или преждевременный отказ от согласованного руководителя требует повторного инициирования процесса и последовательного прохождения всех этапов согласования с самого начала.

Выявленные проблемы носят системный характер и требуют комплексного решения. Отсутствие централизованного учёта обращений приводит к

дублированию информации и росту нагрузки на сотрудников кафедры. Ручное ведение электронных таблиц не гарантирует актуальности данных в реальном времени. Автоматический контроль вакантных мест у руководителей отсутствует, что создаёт риск превышения квоты. Единое информационное пространство не сформировано, что снижает прозрачность распределения студентов. Перечисленные недостатки непосредственно влияют на достижение целевых показателей эффективности, определённых в параграфе 1.3. Устранение этих недостатков является основной задачей разрабатываемой автоматизированной системы. Именно на решение этих проблем и направлен функционал разрабатываемого веб-приложения.

Таблица 4 – Классификация недостатков процесса управления ВКР (AS-IS)

№	Описание недостатка	Характер недостатка	Последствия	Связанное требование
Н1	Студенты выбирают научного руководителя, не имея при этом информации о его текущем количестве вакантных мест на руководство ВКР	Логика и организация процессов	Обращения подаются руководителям, у которых уже нет вакантных мест, что создаёт избыточный поток обращений и неопределённость для студентов	Наличие информации о вакантных местах у руководителей
Н2	Отсутствие автоматического контроля и ограничения количества вакантных мест у руководителей	Логика и организация процессов	Превышение руководителем установленной квоты студентов из-за отсутствия оперативного учёта уже выданных согласий	Автоматический учёт вакантных мест у руководителей на руководство ВКР
Н3	Данные хранятся в разрозненных документах	Информационно-технологический	Разрозненность данных, отсутствие единого достоверного источника информации о закреплении студентов за научными руководителями	Единое информационное пространство
Н4	Отсутствие механизма централизованного отслеживания обращений студентов	Информационно-технологический	Дублирование данных, потеря информации	Единый канал подачи заявок
Н5	Риск ошибок при ручном вводе данных	Организационно-управленческие	Возможность издания некорректных приказов	Достоверность данных

1.5 Выводы

В результате выполнения первой главы настоящей работы проведён анализ предметной области – процесса организации выпускных квалификационных работ в высшем учебном заведении. Приведено описание заказчика, описаны существующие бизнес-процессы и целевая аудитория, включающая четыре группы пользователей.

Проведён обзор и сравнительный анализ существующих программных решений, который показал, что ни одно из рассмотренных решений не обеспечивает комплексную автоматизацию процесса управления ВКР. Обоснована необходимость разработки специализированного веб-приложения.

Определены целевые показатели эффективности разрабатываемой системы. Сформулированы требования к будущей системе. На основе требований выявлены и систематизированы недостатки действующего процесса (AS-IS) с указанием их связи с проектными требованиями.

Проведённый анализ подтвердил наличие организационных и информационно-технологических недостатков существующего процесса управления ВКР, связанных с отсутствием централизованного учёта обращений студентов, автоматического контроля вакантных мест у руководителей и единого информационного пространства.

2 ПРОЕКТИРОВАНИЕ И БИЗНЕС-ЛОГИКА ПРИЛОЖЕНИЯ

2.1 Выбор архитектурного подхода и технологий разработки

В ходе проектирования разработаны диаграммы UML [16, 17] и бизнес-процессы в нотациях BPMN [18, 19] и IDEF0 [20, 21]. На основе этих моделей обоснован выбор технологий для разработки.

В основу разрабатываемой системы положена клиент-серверная архитектура. Взаимодействие между клиентской и серверной частями осуществляется по стандартному протоколу HTTP/HTTPS [22].

Для реализации серверной части выбрана программная платформа Django на языке Python [23–26]. Выбор обусловлен следующими факторами:

- 1) встроенные компоненты для аутентификации, администрирования и работы с базами данных (ORM) ускоряют разработку и сокращают объём шаблонного кода;
- 2) программная платформа обеспечивает защиту от распространённых веб-уязвимостей [27];
- 3) архитектура Django позволяет наращивать функциональность за счёт модульной структуры приложений и обеспечивает возможность дальнейшего масштабирования за счёт добавления кеширования и балансировки нагрузки;
- 4) развёртывание не требует предварительной настройки — достаточно установки зависимостей и применения миграций базы данных;
- 5) проверка работоспособности поддерживается встроенными средствами;
- 6) обширное сообщество и качественная документация облегчают решение возникающих задач.

При выборе инструментов также учитывался современный опыт применения педагогических и информационных технологий в образовании, тенденции цифровой трансформации и методы анализа образовательных данных [28, 29].

В качестве системы управления базами данных на этапе разработки и прототипирования используется СУБД SQLite, которая не требует отдельного серверного процесса и хранит данные в локальном файле, благодаря чему обеспечивается простота локального развёртывания и независимость от дополнительной инфраструктуры [30]. В дальнейшем возможна миграция на промышленную СУБД, например, PostgreSQL, как наиболее распространённое открытое решение для веб-приложений с высокими требованиями к надёжности и производительности [31].

Клиентская часть реализована с использованием стандартных веб-технологий: HTML, CSS и JavaScript [32]. Для стилизации интерфейса применена библиотека стилей Bootstrap, которая позволяет создавать адаптивный и современный дизайн [33].

Для решения специфических задач привлечены следующие библиотеки языка программирования Python:

- 1) Pandas и Openpyxl для формирования и экспорта реестров заявок в формате Excel [34];
- 2) Python-docx для программного заполнения шаблонов и генерации проектов приказов в формате Word [35].

Выбранный стек технологий полностью покрывает функциональные требования системы на всех этапах её работы.

2.2 Формализация требований и проектирование модели данных

На основе анализа предметной области и существующего процесса, проведённого в первой главе настоящей работы, сформулированы следующие функциональные требования к системе, сгруппированные по ролям пользователей.

Для студента:

- 1) просмотр списка руководителей своей кафедры с отображением области научных интересов и количества вакантных мест на руководство ВКР;

2) подача заявки на руководство ВКР с указанием выбранного руководителя и темы работы;

3) просмотр поданных заявок и их статусов;

4) просмотр истории своей заявки;

5) отзыв заявки;

6) восстановление отозванной заявки;

7) редактирование темы заявки.

Для руководителя:

1) просмотр списка входящих заявок от студентов;

2) одобрение или отклонение заявок;

3) просмотр количества вакантных мест на руководство ВКР;

4) отзыв ранее одобренной заявки;

5) восстановление отозванной заявки;

6) подача запроса на изменение квоты.

Для заведующего кафедрой:

1) просмотр списков руководителей, студентов и заявок в рамках своей кафедры;

2) просмотр справочников в рамках своей кафедры;

3) обработка запросов руководителей на изменение квоты;

4) формирование реестра заявок в формате Excel;

5) формирование проекта приказа о назначении руководителей и утверждении тем выпускных квалификационных работ.

Для администратора: управление справочниками.

Нефункциональные требования:

1) время отклика на стандартные операции не более 2 секунд при одновременной работе до 100 пользователей;

2) доступность системы 24/7 в период подачи заявок;

3) разграничение прав доступа на основе ролей, включая ограничение доступа пользователей данными своей кафедры;

4) возможность расширения функциональности за счёт модульной структуры программной платформы Django.

Кроме того, системные требования ограничивают количество активных заявок от одного студента тремя. Активными считаются заявки со статусами «Ожидает» и «Одобрена». Отклонённые и отозванные заявки при подсчёте не учитываются.

Представленный перечень требований охватывает все ключевые роли и отражает специфику процесса управления ВКР, выявленную в ходе анализа предметной области. Особенностью сформулированных требований является наличие механизмов, отсутствующих в типовых образовательных системах: двусторонний отзыв и восстановление заявок, автоматический учёт вакантных мест у руководителей на руководство ВКР, а также возможность запроса изменения квоты руководителями. Эти функции в совокупности формируют основу для гибкого и прозрачного процесса распределения студентов по научным руководителям.

Взаимодействие пользователей с системой представлено на рисунке А.2.

В качестве следующего шага проектирования была разработана инфологическая модель базы данных (рисунок А.3), описывающая ключевые сущности системы и связи между ними. В число основных логических блоков модели входят: «Пользователь», «Студент», «Руководитель», «Заявка», «Запрос на изменение квоты», а также набор справочных сущностей, а именно: факультеты, кафедры, образовательные профили, должности, учёные степени, учёные звания, категории технологий и сами технологии. Такая структура позволяет зафиксировать все необходимые для работы системы данные и отношения между ними на концептуальном уровне [36–38].

Переход к даталогической модели позволяет задать типы данных для каждого атрибута сущности, сохраняя независимость логики от технической реализации хранения.

Между основными сущностями установлены следующие ключевые связи:

1) отношение «один к одному» между «Пользователем» и «Студентом», а также между «Пользователем» и «Руководителем» отражает специализацию учётной записи;

2) отношения «один ко многим» от «Студента» к «Заявкам» и от «Руководителя» к «Заявкам». Эти две связи совместно реализуют отношение «многие ко многим» между студентами и руководителями, опосредованное через сущность «Заявка»;

3) отношение «один ко многим» между сущностями «Руководитель» и «Запрос на изменение квоты».

Спроектированная структура базы данных позволяет корректно реализовать ключевую бизнес-логику: автоматический контроль вакантных мест руководителя, ведение истории изменений каждой заявки и поддержку механизмов отзыва и восстановления.

Таким образом, разработанная модель данных в совокупности с ролевой моделью образует целостную проектную основу для технической реализации системы.

На основе концептуальной (инфологической) модели была разработана даталогическая модель (рисунок А.4), определяющая детальную структуру таблиц, их атрибуты, типы данных, первичные и внешние ключи для реализации в СУБД SQLite. Ключевые таблицы базы данных:

1) модель user отвечает за хранение учётных данных, реализована встроенной моделью User Django;

2) модель student содержит профиль студента, связана с user отношением «один к одному» через user_id;

3) модель professor содержит профиль руководителя, связана с user отношением «один к одному» через user_id;

4) модель thesisapplication представляет собой центральную таблицу заявок, включает ссылки на студента и руководителя, тему, статус, журнал изменений в формате JSON, а также временные метки;

5) модель `quotachangerequest` хранит запросы руководителей на изменение количества вакантных мест, включает ссылку на руководителя, текущие и запрашиваемые значения, статус запроса и обоснование;

6) модель `application_technology` обеспечивает связь многие ко многим между заявками и технологиями.

Справочные таблицы (`faculty`, `department`, `profile_education`, `position`, `degree`, `rank`, `technology`, `technology_category`) обеспечивают гибкую настройку системы без изменения программного кода.

В системе определены четыре роли пользователей с соответствующими правами доступа (таблица Б.2).

В системе предусмотрены две роли с доступом в административную панель: заведующий кафедрой (`is_staff=True`, `is_superuser=False`) и администратор (`is_superuser=True`). Заведующий кафедрой видит и может редактировать данные только своей кафедры. Администратор обладает полными правами и отвечает за ведение справочников. Суперпользователь, создаваемый через команду `createsuperuser`, неотличим от администратора и используется на этапе внедрения. Разделение ролей `is_staff` и `is_superuser` через средства административной панели Django позволяет гибко настраивать права доступа без изменения программного кода. Таким образом, в системе может быть создана учётная запись с правами «заведующий кафедрой» не только для заведующего, но и для иного сотрудника, например, для руководителя образовательной программы, которому делегированы функции по формированию приказов или обработке запросов.

В соответствии с теоретико-множественным подходом [39] разрабатываемая система представляется в виде кортежа из трёх базовых множеств:

$$S = \langle U, R, A \rangle, \quad (1)$$

где $U = \{u_1, u_2, \dots, u_n\}$ — множество пользователей системы,

$R = \{r_1, r_2, r_3, r_4\}$ — множество ролей,

$A = \{a_1, a_2, \dots, a_m\}$ — множество заявок на руководство ВКР.

Принадлежность пользователя к роли задаётся функцией:

$$\rho: U \rightarrow R. \quad (2)$$

Описанное в таблице Б.2 разграничение прав доступа формально задаётся функцией (2).

Каждой заявке в конкретный момент времени соответствует один из допустимых статусов. Жизненный цикл заявки описывается функцией перехода между статусами:

$$\sigma: A \rightarrow \{\text{«Ожидает»}, \text{«Одобрена»}, \text{«Отклонена»}, \text{«Отозвана»}\}. \quad (3)$$

В контексте формализации рассматриваемой предметной области применение кортежа (1) обусловлено логикой самого процесса: сначала в системе появляются пользователи, затем им назначаются роли, определяющие их права, и только после этого возникают заявки как результат взаимодействия пользователей с разными ролями. Порядок перечисления компонентов кортежа отражает эту причинно-следственную связь.

Формализация используется для описания ролевой структуры системы и жизненного цикла заявки.

Аналогичный подход к структурированию системы через кортеж множеств использован в исследовании диагностики рисков подготовки выпускных квалификационных работ [40], в которой адаптация метафоры «атакующая поверхность» из информационной безопасности к образовательной среде позволяет представить процесс подготовки ВКР как систему компонентов, уязвимостей и негативных исходов.

2.3 Описание бизнес-процессов и бизнес-логики

На основе сформулированных в первой главе требований был спроектирован целевой бизнес-процесс взаимодействия пользователей с системой. Его ядром является жизненный цикл заявки (рисунок 3).

Система предоставляет студенту актуальный список руководителей со своей кафедры с указанием вакантных мест на руководство ВКР. Ключевым элементом процесса является автоматическая проверка наличия вакантных мест руководителя в момент выбора: если все места заняты, система выводит соответствующее сообщение, в противном случае открывается форма для создания заявки.

После подачи заявки она поступает на рассмотрение выбранному руководителю. Руководитель может одобрить заявку или отклонить её. Одобрение заявки автоматически уменьшает счётчик вакантных мест руководителя на единицу. Студент также может отозвать свою заявку на любом этапе. Если на момент отзыва заявка была одобрена, счётчик вакантных мест руководителя автоматически увеличивается на единицу.

Отозванная заявка может быть восстановлена только той стороной, которая её отозвала.

Все операции с заявками фиксируются в журнале изменений, что обеспечивает прослеживаемость и аудит. Итоговым административным процессом является формирование проекта приказа на основе всех одобренных заявок по выбранной учебной группе.



Рисунок 3 – Основные этапы жизненного цикла заявки

Диаграмма процесса «Подача заявки в системе студентом» в нотации BPMN 2.0 (TO-BE) представлена на рисунке А.5.

Отдельным процессом является запрос руководителем изменения квоты. Руководитель заполняет форму с указанием желаемого количества мест и обоснованием. Запрос поступает на рассмотрение заведующему кафедрой, который может одобрить его, тогда квота изменяется, или отклонить.

На рисунке А.6 представлена диаграмма последовательности для сценариев одобрения и отклонения запроса на изменение квоты.

Целевой бизнес-процесс спроектирован таким образом, чтобы максимально исключить ручное вмешательство и повысить скорость обработки заявок. Он охватывает полный жизненный цикл заявки: от выбора руководителя студентом до формирования проекта приказа заведующим кафедрой. Ключевыми преимуществами спроектированного бизнес-процесса являются автоматическая проверка вакантных мест руководителя на руководство ВКР, двусторонний отзыв и восстановление заявок, а также механизм запроса изменения квоты.

Дополнительно к BPMN для описания системы на обобщённом уровне была разработана функциональная модель IDEF0 (TO-BE). На приведённой диаграмме указаны:

- 1) входы, коими являются заявки на руководство ВКР от студентов, выбранные заведующими кафедр учебные группы для формирования проекта приказа и запросы на изменение квоты от руководителей;
- 2) управления, коими являются положения о ВКР, квалификационный справочник, номенклатура направлений, квоты и шаблон приказа;
- 3) механизмы, коими являются студенты, руководители, заведующие кафедрами и администраторы;
- 4) выходы, коими являются проекты приказов в формате Word, реестры с заявками в формате Excel, обновлённые справочники, обновлённые статусы заявок, обновлённые квоты.

Диаграмма IDEF0 (TO-BE) и её декомпозиция представлены на рисунках А.7 и А.8 соответственно.

Разработанные сценарии для ключевых процессов «Подача заявки студентом», «Одобрение заявки руководителем» и «Формирование проекта приказа заведующим кафедрой» представлены в таблицах Б.3, Б.4 и Б.5 соответственно.

Представленные сценарии охватывают как штатные ситуации, так и альтернативные потоки, включая отсутствие вакантных мест, дублирование заявок и отказы. Такая детализация позволяет проверить корректность работы системы при различных условиях. Описанные сценарии легли в основу функционального тестирования (параграф 3.3), в ходе которого подтверждена работоспособность всех ключевых модулей.

Требования к системе также формализованы в виде пользовательских историй:

- 1) «Как студент, я хочу видеть количество вакантных мест у каждого руководителя, чтобы выбрать того, кто ещё может взять студентов на руководство ВКР»;
- 2) «Как студент, я хочу видеть область научных интересов руководителя, чтобы выбрать того, чья тематика мне ближе»;
- 3) «Как студент, я хочу иметь возможность отозвать заявку, если мои обстоятельства изменились»;
- 4) «Как студент, я хочу иметь возможность восстановить отозванную заявку, чтобы не подавать её заново»;
- 5) «Как руководитель, я хочу видеть все входящие заявки в одном месте, чтобы централизованно управлять ими»;
- 6) «Как руководитель, я хочу знать, сколько вакантных мест на руководство ВКР у меня есть»;
- 7) «Как руководитель, я хочу отправить запрос на изменение квоты, если мои обстоятельства изменились»;

8) «Как руководитель, я хочу иметь возможность отозвать ранее одобренную заявку, если мои обстоятельства изменились»;

9) «Как заведующий кафедрой, я хочу просматривать данные своей кафедры и формировать проекты приказов по группам»;

10) «Как администратор, я хочу иметь возможность добавлять, удалять, редактировать записи в справочниках без изменения программного кода».

Основные правила и алгоритмы системы, реализующие ключевые бизнес-процессы, формализованы ниже.

Проверка наличия вакантных мест у руководителя перед созданием заявки:

1) если текущее количество закреплённых за руководителем студентов меньше его квоты на руководство ВКР, заявка может быть создана;

2) если текущее количество закреплённых студентов равно или превышает квоту, система блокирует создание заявки и выводит соответствующее сообщение.

Обновление количества занятых мест при изменении статуса заявки:

1) при одобрении заявки количество студентов, закреплённых за руководителем на руководство ВКР, увеличивается на единицу;

2) при отзыве одобренной заявки количество закреплённых студентов уменьшается на единицу;

3) при отклонении заявки количество закреплённых студентов не изменяется;

4) при восстановлении заявки, которая на момент отзыва имела статус «Одобрена», количество закреплённых студентов увеличивается на единицу.

Запрос изменения квоты:

1) руководитель создаёт запрос, указывая желаемую квоту на руководство ВКР и обоснование;

2) при одобрении запроса заведующим кафедрой значение квоты обновляется;

3) при отклонении запроса заведующим кафедрой значение квоты не изменяется.

Формирование проекта приказа:

- 1) проект приказа формируется на основе всех заявок со статусом «Одобрена» в выбранной учебной группе;
- 2) данные подставляются в шаблон документа Word с использованием библиотеки python-docx.

При каждом изменении статуса заявки, редактировании темы или других действиях в поле history добавляется новая запись в формате JSON, содержащая временные метки, имя и роль пользователя, тип действия и детали изменения.

Приведённые выше правила формализуют логику работы системы. Так, критерий «руководитель имеет вакантные места» реализуется через сравнение атрибутов `current_load` и `max_load` модели Professor. Событие «отзыв заявки студентом» реализуется как изменение атрибута `status` со значения «Ожидает» или «Одобрена» на значение «Отозвана» с сохранением предыдущего статуса в поле `previous_status` для возможности последующего восстановления.

На рисунке 4 представлены возможные статусы заявки, формально описываемые функцией (3) в параграфе 2.2.

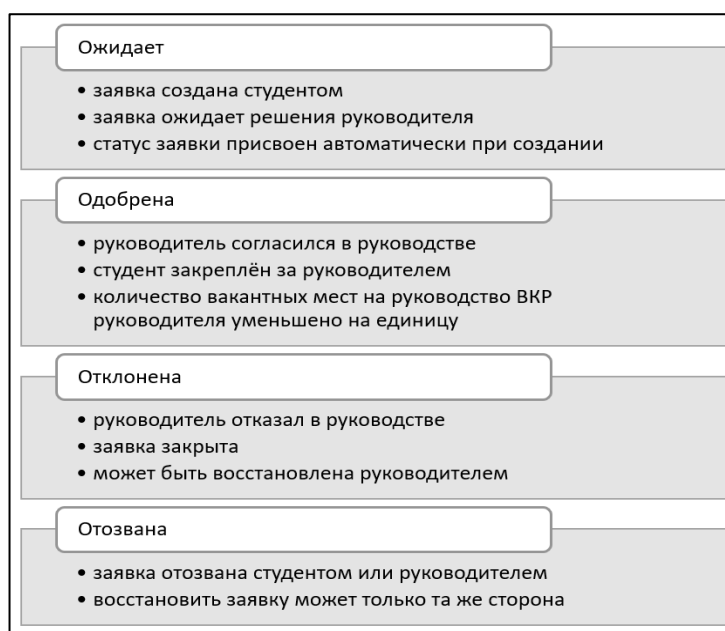


Рисунок 4 – Статусы заявки и их описания.

Диаграмма процесса проверки наличия вакантных мест у руководителя представлена на рисунке 5 в нотации EPC [41, 42].

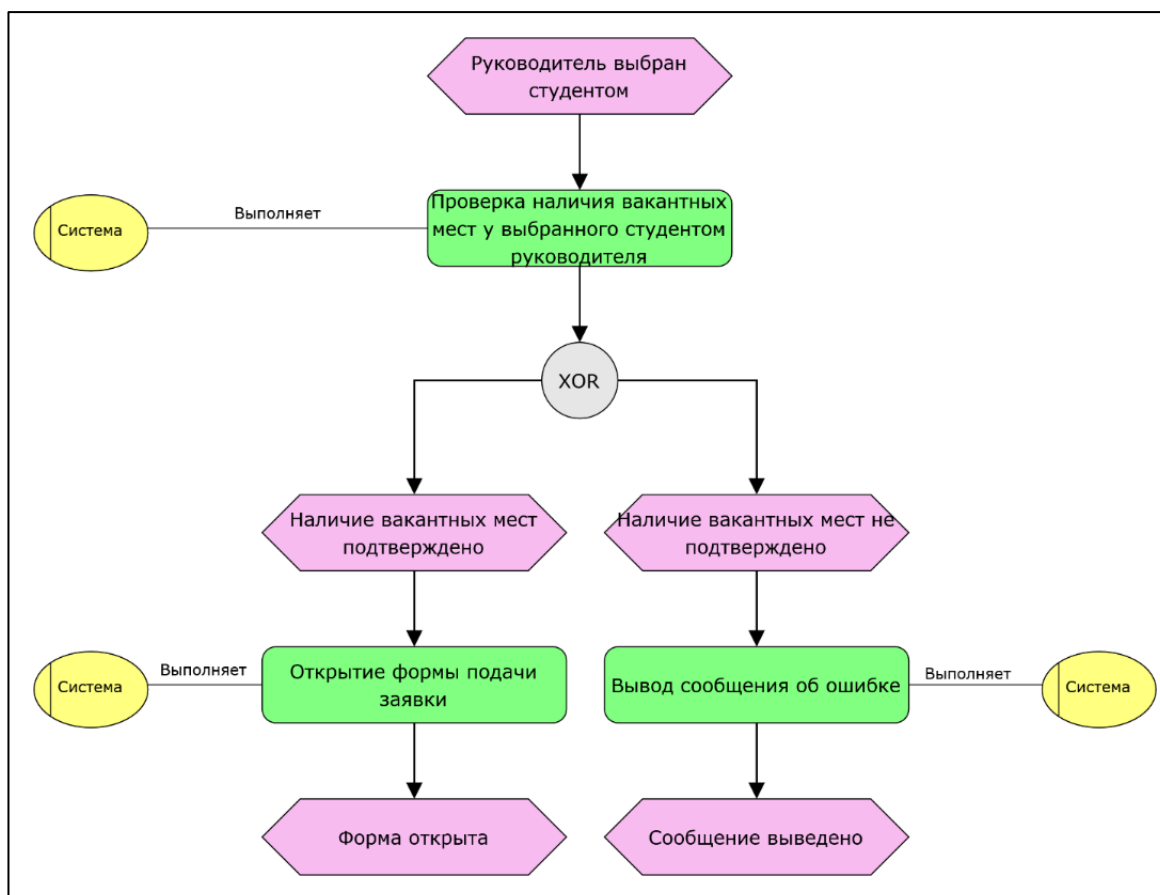


Рисунок 5 – Диаграмма процесса проверки наличия вакантных мест у руководителя в нотации EPC (TO-BE)

2.4 Проектирование пользовательского интерфейса

Проектирование пользовательского интерфейса выполнялось непосредственно в процессе разработки веб-приложения с использованием библиотеки стилей Bootstrap для стилизации и адаптивной вёрстки. Данная библиотека стилей предоставляет готовые адаптивные компоненты. Это позволило создать интерфейс, корректно отображающийся на различных устройствах. Итеративный подход к разработке в сочетании с обратной связью от целевой аудитории позволил своевременно уточнять функциональность и улучшать интерфейс.

При проектировании интерфейса учитывались требования к эргономике интерфейса и принципы человеко-ориентированного проектирования, использовались принципы минимизации количества действий пользователя, единообразия элементов управления и визуального выделения критически важной информации [43, 44].

Структура навигации веб-приложения строится на основе ролевой модели, описанной в параграфе 2.2. После авторизации пользователю доступен личный кабинет, содержимое которого зависит от его роли.

Основные разделы системы приведены ниже.

Для студента:

- 1) список руководителей с указанием количества вакантных мест на руководство ВКР и области научных интересов;
- 2) форма подачи заявки с возможностью указания темы, описания и используемых технологий;
- 3) страница с отображением статусов поданных заявок и кнопками отзыва/восстановления.

Для руководителя:

- 1) список входящих заявок с возможностью фильтрации по статусам;
- 2) строка с информацией о количестве вакантных мест;
- 3) форма запроса изменения квоты.

Для заведующего кафедрой:

- 1) административная панель с данными по своей кафедре;
- 2) просмотр списка руководителей и студентов своей кафедры;
- 3) раздел обработки запросов на изменение квот руководителей;
- 4) инструменты экспорта реестров и генерации проектов приказов;
- 5) просмотр информации о вакантных местах по каждому руководителю.

Для администратора: панель для управления справочниками системы.

Скриншоты ключевых экранов, демонстрирующие функциональность и пользовательский интерфейс системы, представлены на рисунках А.9–А.17.

Сформированный проект приказа и экспортированный реестр заявок приведены на рисунках А.18–А.19 и А.20 соответственно.

2.5 Выводы

В результате выполнения второй главы разработана проектная модель веб-приложения.

Обоснован выбор архитектурного подхода и технологического стека.

Выполнено проектирование информационной системы: сформулированы функциональные требования для четырёх ролей пользователей, разработаны диаграммы UML (прецедентов и последовательностей), инфологическая и даталогическая модели базы данных, а также матрица ролей и прав доступа.

Выполнена формализация системы в виде кортежа множеств (1), функций принадлежности пользователя к роли (2) и перехода между статусами заявки (3), что позволило строго описать ролевую модель и жизненный цикл заявки.

Спроектированы целевые бизнес-процессы (TO-BE) в нотациях BPMN 2.0 и IDEF0, описывающие жизненный цикл заявки, процессы отзыва, восстановления, запроса изменения квоты и формирования проекта приказа. Разработаны сценарии использования (Use Cases) и пользовательские истории (User Stories). Формализована ключевая бизнес-логика: правила проверки наличия вакантных мест у руководителя, обновления количества вакантных мест при изменении статуса заявки, обработки запросов на изменение квоты, формирования проекта приказа и ведения журнала изменений.

Выполнено проектирование пользовательского интерфейса с учётом ролевой модели.

Результаты второй главы создают проектную основу для реализации веб-приложения, которая описана в третьей главе.

3 ТЕХНИЧЕСКАЯ РЕАЛИЗАЦИЯ И ОЦЕНКА ЭФФЕКТИВНОСТИ

3.1 Реализация серверной части

Серверная часть веб-приложения реализована на языке Python 3.12 с использованием программной платформы Django 4.2. Проект организован по модульному принципу [45] и включает пять приложений:

- 1) приложение accounts отвечает за управление пользователями, профилями студентов и руководителей, запросами на изменение квоты;
- 2) приложение applications отвечает за управление заявками на ВКР, генерацию проектов приказов и экспорт реестров;
- 3) приложение core отвечает за общие настройки, маршрутизацию и главную страницу;
- 4) приложение references отвечает за справочники, а именно факультеты, кафедры, профили, должности, учёные степени и звания;
- 5) приложение technologies отвечает за справочник технологий, указываемых студентом при подаче заявки.

Центральным элементом серверной части является модель User, расширяющая стандартную модель Django (AbstractUser [46]). В неё добавлено поле role, которое определяет роль пользователя в системе: «студент», «руководитель» или «администратор». Роль «заведующий кафедрой» реализована не через отдельное значение поля role, а за счёт включения в группу «Заведующий кафедрой», что позволяет гибко назначать и снимать соответствующие права без изменения модели.

С профилем пользователя связаны две модели: Student и Professor. Каждая из них соединена с User отношением «один к одному». Модель Student содержит поля: full_name, group, faculty, department, profile, а также контактные данные (поля email, phone). Модель Professor, помимо контактных данных и принадлежности к кафедре, включает поля current_load (текущее количество занятых мест на руководство ВКР), max_load (квота), а также ссылки на должность, учёную степень и учёное звание. Кроме того, модель Professor содержит

текстовое поле `research_interests` для указания руководителем своей области научных интересов. Эта информация отображается в списке руководителей для студентов при выборе руководителя.

Бизнес-логика работы с заявками сосредоточена в приложении `applications`. Модель `ThesisApplication` описывает заявку (листинг В.1) и содержит следующие атрибуты:

- 1) поля `student` и `professor` – внешние ключи на соответствующие модели;
- 2) поле `topic` – тема выпускной квалификационной работы;
- 3) поле `description` – описание работы;
- 4) поле `technologies` – связь «многие ко многим» со справочником технологий;
- 5) поле `technologies_other` – текстовое поле для технологий, отсутствующих в справочнике;
- 6) поле `status` – статус заявки;
- 7) поля `previous_status` и `withdrawn_by` – поля, используемые при отзыве заявки для возможности последующего восстановления;
- 8) поле `history` – журнал изменений в формате JSON, хранящий историю всех операций с заявкой.

Для обеспечения целостности данных при изменении статусов заявок используются сигналы Django [47]. Функция-обработчик `update_professor_load_on_save`, вызываемая при сохранении заявки, отслеживает изменение статуса и автоматически обновляет счётчик вакантных мест руководителя: при одобрении заявки счётчик уменьшается на единицу, при отзыве одобренной заявки — увеличивается на единицу. Для исключения ошибок при одновременной работе нескольких пользователей обновление счётчика вакантных мест выполняется с проверкой текущего значения и блокировкой записи на время операции.

Механизм запроса изменения квоты руководителя реализован через модель QuotaChangeRequest. Руководитель через личный кабинет отправляет запрос, указывая желаемое значение квоты и обоснование. Запрос сохраняется в базе данных со статусом «Ожидает рассмотрения». Заведующий кафедрой через административную панель может одобрить запрос – тогда значение max_load руководителя обновляется – или отклонить его.

Для предотвращения нецелевого использования системы введено ограничение на количество активных заявок от одного студента. Активными считаются заявки со статусами «Ожидает» и «Одобрена». Система ограничивает количество активных заявок от одного студента тремя. При достижении этого лимита попытка подать новую заявку блокируется с выводом соответствующего сообщения об ошибке. Отклонённые и отозванные заявки не учитываются при подсчёте, что позволяет студенту освободить лимит при необходимости. Реализация данного ограничения выполняется на уровне представления при создании новой заявки. Это исключает превышение лимита даже при прямых HTTP-запросах. Фрагмент программного кода, демонстрирующий эту логику, приведён в листинге 3.1.

Листинг 3.1 – Фрагмент функции create_application представления системного приложения applications

```
1 ...
2 max_active = 3
3 if active_count >= max_active:
4     messages.error(
5         request,
6         f'Вы не можете иметь более {max_active} активных заявок.'
7     )
8 ...
```

В документ последовательно добавляются шапка с реквизитами университета, заголовок «ПРИКАЗ», дата и номер, заголовок «О назначении руководителей и утверждении тем выпускных квалификационных работ», преамбула

с указанием направления подготовки, распорядительная часть с таблицей студентов (ФИО, тема, руководитель), основание для издания приказа и реквизиты декана факультета.

При формировании документа применяются параметры форматирования, соответствующие утверждённому шаблону приказа университета.

Документ формируется в памяти сервера и передаётся пользователю для скачивания.

Экспорт реестра заявок в формат Excel реализован в методе `export_registry`. С использованием библиотек `pandas` и `openpyxl` формируется таблица, содержащая следующие столбцы: ФИО студента, группа, тема ВКР, научный руководитель, кафедра, статус, дата создания. Файл передаётся пользователю для скачивания.

Административная панель построена на основе встроенной программной платформы администрирования Django [48]. Для каждой модели зарегистрирован соответствующий административный класс [49], определяющий:

- 1) поля, отображаемые в списке объектов (`list_display`);
- 2) фильтры для быстрого поиска (`list_filter`);
- 3) поля, доступные только для чтения (`readonly_fields`);
- 4) группировку полей в форме редактирования (`fieldset`).

Для роли «заведующий кафедрой» реализована фильтрация данных: через переопределение метода `get_queryset` административных классов заведующий кафедрой видит только студентов, руководителей и заявки со своей кафедры. Права доступа назначены через группу «Заведующий кафедрой» с использованием системы разрешений программной платформы Django.

Взаимодействие с базой данных осуществляется через ORM Django, что обеспечивает независимость от конкретной СУБД. На этапе разработки используется SQLite, однако при необходимости переноса системы в промышленную эксплуатацию поддерживается миграция на промышленную СУБД.

3.2 Разработка клиентской части

Клиентская часть веб-приложения реализована на HTML-шаблонах Django Template Language (DTL) [50] с применением библиотеки стилей Bootstrap 5, обеспечивающего адаптивную вёрстку и единообразный внешний вид всех страниц. Шаблоны построены на основе базового шаблона `base.html`, содержащего заголовочную и нижнюю части веб-приложения.

Для каждой роли пользователя разработан собственный набор страниц, реализующий функции, перечисленные в параграфе 2.2. Все формы используют CSRF-токен – встроенную защиту Django от межсайтовой подделки запросов [51]. Результаты действий пользователя отображаются через систему сообщений Django: успешные операции подсвечиваются зелёным фоном, ошибки — красным. Визуальная обратная связь через цветное выделение сообщений повышает удобство работы пользователя и снижает риск неверной интерпретации результатов действий.

Для динамической загрузки списка групп при формировании приказа реализованы две вспомогательные функции, возвращающие данные в формате JSON.

Обе функции выполняют запрос к базе данных и возвращают результат:

1. `/api/groups-list/` возвращает список всех учебных групп.
2. `/api/groups-with-approved/` возвращает список групп, в которых есть хотя бы одна одобренная заявка.

Такой подход позволяет обновлять содержимое выпадающих списков и других динамических элементов без полной перезагрузки страницы, что ускоряет работу пользователя. Для остальных данных используется стандартная передача через шаблоны Django.

В системе реализован механизм модальных окон для просмотра контактных данных пользователей. При нажатии на имя студента отображаются его учебная группа, электронная почта и телефон, при нажатии на имя руководителя – кафедра, электронная почта и телефон. Клик по теме заявки открывает

окно с детальной информацией: тема, описание, использованные технологии, статус. Отдельное модальное окно отображает полную историю изменений заявки с отображением действия, времени, пользователя. Каждое из этих окон оснащено функцией печати, позволяющей вывести содержимое на принтер или сохранить в формате PDF с помощью штатных средств браузера.

3.3 Тестирование и направления для развития

Основным видом тестирования выступило приёмочное тестирование [52–55], проведённое методом ручной проверки пользовательских сценариев на работающем экземпляре системы. Для каждой роли были выделены ключевые сценарии, охватывающие основной жизненный цикл заявки и смежные процессы. Тест-кейсы разработаны на основе сценариев использования (Use Cases), представленных в параграфе 2.3. Результаты тестирования представлены в таблице Б.6.

Всего было разработано и успешно выполнено 13 тест-кейсов, охватывающих 4 роли пользователей и 5 ключевых бизнес-сценариев. Проверке подверглись все функциональные требования, перечисленные в таблице 4, что позволило подтвердить корректность работы системы по всем заявленным сценариям.

Успешное выполнение всех 13 тест-кейсов подтверждает корректность реализации заявленных функциональных требований. Полученные результаты тестирования и выполненная экспертная оценка позволяют сделать вывод о достижении целевых показателей, сформулированных в параграфе 1.3.

Для наполнения системы тестовыми данными была применена локально запущенная нейросетевая модель Qwen2.5 7B Instruct через утилиту Ollama [56]. На основе первичного запроса (Приложение Г) и последующего диалога, в ходе которого модель знакомилась с архитектурой проекта и структурой базы данных, был сгенерирован исполняемый программный код на Python, который был адаптирован и интегрирован в проект для имитации реального рассматриваемого процесса и наполнения базы данных тестовыми записями.

Применение большой языковой модели позволило быстро сгенерировать реалистичный набор тестовых данных, ускорив процесс тестирования на начальных этапах разработки.

Учитывая стабильную работу ядра системы, можно выделить следующие основные направления для дальнейшего развития:

1. Интеграция с архитектурой Московского политехнического университета, которая позволит пользователям входить в систему, используя корпоративную учётную запись. Это позволит исключить необходимость хранения паролей или иных аутентификационных данных в базе данных системы.

2. Внедрение уведомлений по электронной почте для информирования пользователей об изменении статуса заявок.

3. Расширение функциональности формирования приказов, добавляющее возможность формирования других типов документов.

4. Переход на промышленную СУБД (например, PostgreSQL) для повышения производительности и надёжности при увеличении пользовательской нагрузки, а также для поддержки более сложных запросов и обеспечения целостности данных на уровне сервера баз данных.

5. Предоставление заведующему кафедрой возможности конфигурирования максимального количества активных заявок для студентов своей кафедры непосредственно через интерфейс административной панели без модификации программного кода.

6. Реализация модуля планирования консультаций, позволяющего научному руководителю назначать время, место и формат встреч с закреплёнными за ним студентами непосредственно в системе.

Отдельным перспективным направлением, заслуживающим внимания, является реализация поддержки формата «Стартап как диплом», где выпускная квалификационная работа может выполняться командой из нескольких студентов, совместно разрабатывающих бизнес-проект [57]. Актуальна также и адаптация системы под межкафедральное руководство, когда научный руководитель назначается от двух кафедр, а не одной, что расширяет для студента

выбор доступных научных руководителей за счёт руководителей с разных кафедр. Примером межкафедрального руководства служит образовательная программа «Информационные технологии управления бизнесом» Московского политехнического университета, объединяющая технологическую и управленческую составляющие [58].

Программный код проекта приведён к единому стилю в соответствии со стандартами оформления PEP8 с использованием инструментов Black и isort, что обеспечивает единообразие форматирования и облегчает поддержку кода. Отдельные предупреждения утилиты проверки стиля Flake8 о длине строк и расположении импортов в служебных файлах были оставлены без изменений, так как они не влияют на читаемость и функциональность кода, характерны для типовых Django-проектов и не нарушают требований к промышленной разработке.

3.4 Оценка эффективности разработанной системы

Оценка эффективности разработанного веб-приложения выполнена путём сравнительного анализа ключевых показателей процесса управления ВКР (AS-IS) и проектных значений (TO-BE). Базовые показатели определены на основе наблюдения за действующим процессом и экспертных оценок сотрудников кафедры; проектные значения получены путём выполнения операций в разработанной системе.

Обобщённые результаты сравнительного анализа по всем четырём критериям представлены в таблице 5.

Таблица 5 – Сравнительный анализ показателей эффективности

Критерий	Значение AS-IS	Значение TO-BE	Эффект
Время выбора и закрепления руководителя	До 5 суток	Менее 1 суток	Сокращение в 5 раз
Трудоёмкость формирования одного приказа	4 рабочих часа	0,2 рабочих часа	Сокращение в 20 раз
Доля ошибок при вводе данных	10-15%	До 1%	Снижение до 1%
Время получения сводки о ходе распределения студентов	300 минут	До 1 минуты	Сокращение с часов до секунд

Таким образом, использование разработанной системы позволяет существенно сократить временные и трудовые затраты, повысить достоверность данных и обеспечить информированность участников процесса.

3.5 Выводы

В результате выполнения третьей главы настоящей работы осуществлена техническая реализация веб-приложения для автоматизации управления выпускными квалификационными работами и выполнена оценка его эффективности.

Серверная часть реализована на базе программной платформы Django с использованием пяти приложений. Реализована расширенная модель пользователя с ролевым разграничением, включая роль заведующего кафедрой на основе встроенных групп пользователей Django. Бизнес-логика обновления счётчика вакантных мест при изменении статуса заявки реализована через сигналы, что обеспечивает целостность данных. Механизм запроса изменения квоты руководителя, генерация проектов приказов и экспорт реестров реализованы с использованием библиотек `python-docx`, `pandas` и `openpyxl`.

Клиентская часть представляет собой набор HTML-страниц, реализованных с применением библиотеки стилей Bootstrap 5. Интерфейс адаптирован под четыре роли пользователей. Для динамической загрузки списка учебных групп при формировании приказа реализованы две вспомогательные функции, возвращающие данные в формате JSON.

Проведено ручное функциональное тестирование системы на локальном сервере. Разработано 13 тест-кейсов, охватывающих критические пути всех ролей пользователей: от подачи заявки студентом до формирования проекта приказа заведующим кафедрой. Все тест-кейсы выполнены успешно, система функционирует в соответствии с заявленными требованиями.

Проведённая оценка эффективности показывает, что разработанная система обеспечивает достижение всех установленных целевых показателей, определённых в параграфе 1.3.

Определены направления для дальнейшего развития.

Разработанная база данных содержит 16 таблиц: 5 основных сущностей, 2 служебные таблицы Django, 8 справочников и 1 связочную таблицу.

Общий объём написанного программного кода составляет 4969 строк, из которых 87,1% написано автором, а 12,9% сгенерировано нейросетевой моделью Qwen2.5 7B Instruct и адаптировано под требования проекта.

ЗАКЛЮЧЕНИЕ

В результате выполнения выпускной квалификационной работы была достигнута поставленная цель и решены все поставленные задачи. Разработано веб-приложение для автоматизации управления выпускными квалификационными работами. Проведённое тестирование и оценка эффективности позволяют сделать вывод о достижении целевых показателей, сформулированных в работе: система обеспечивает сокращение времени выбора научного руководителя до 5 раз, снижение трудоёмкости формирования приказов до 20 раз и уменьшение доли ошибок с 10-15% до 1%, что доказывает её практическую ценность и обосновывает возможность использования в деятельности кафедры.

На этапе анализа предметной области было приведено описание заказчика – кафедры «Инфокогнитивные технологии» Московского политехнического университета, описан существующий процесс управления ВКР (AS-IS) и определены четыре группы пользователей: студенты, руководители, заведующие кафедрами и администраторы. Проведённый обзор аналогов показал, что ни одно из существующих решений не обеспечивает полной автоматизации рассматриваемого процесса. На основе сформулированных требований выявлены и классифицированы недостатки действующего процесса, определены целевые показатели эффективности (TO-BE) по методике SMART.

На этапе проектирования обоснован выбор архитектурного подхода и технологического стека, разработаны диаграммы UML (прецедентов и последовательностей), инфологическая и даталогическая модели базы данных в нотации Мартина, а также матрица ролей и прав доступа для четырёх категорий пользователей. Спроектированы целевые бизнес-процессы в нотациях BPMN 2.0 и IDEF0, описывающие жизненный цикл заявки, процессы отзыва, восстановления, запроса изменения квоты и формирования проекта приказа. Разработаны сценарии использования и пользовательские истории, формализована ключевая бизнес-логика.

Математическая формализация системы (1)–(3) позволила формально описать ключевые сущности и правила функционирования системы, что обеспечило логическую целостность проектных решений и корректность реализации бизнес-процессов.

На этапе реализации создана серверная часть на базе программной платформы Django, включающая пять приложений, реализованы модели данных, сигналы для автоматического обновления счётчика вакантных мест, механизм запроса изменения квоты руководителя, генерация проектов приказов в формате Word с оформлением по шаблону университета и экспорт реестров в формате Excel. Клиентская часть построена на шаблонах Django с применением библиотеки стилей Bootstrap 5, интерфейс адаптирован под четыре роли пользователей. Для загрузки данных при формировании приказа реализованы вспомогательные функции, возвращающие данные в формате JSON.

Проведено приёмочное тестирование методом ручной проверки пользовательских сценариев. Разработано 13 тест-кейсов, охватывающих критические пути всех ролей пользователей. Все тест-кейсы выполнены успешно, что подтверждает соответствие системы заявленным требованиям.

Сравнительный анализ показателей эффективности показал, что использование разработанной системы позволяет сократить время выбора и закрепления научного руководителя до 5 раз, трудоёмкость формирования проекта приказа – до 20 раз, долю ошибок при вводе данных – с 10–15% до 1%, а время получения сводки о ходе распределения студентов – с нескольких часов до десятков секунд.

Таким образом, целевые показатели эффективности, определённые в параграфе 1.3, достигнуты, что подтверждает практическую ценность системы.

Практическая ценность работы заключается в возможности использования разработанного приложения на кафедре или факультете для автоматизации процесса управления ВКР, что позволяет повысить информированность участников рассматриваемого процесса о текущем распределении студентов выпускных курсов по научным руководителям и достоверность данных.

Техническое задание на разработку приведено в Приложении Д.

Определены направления для дальнейшего развития: интеграция с архитектурой университета для использования корпоративной учётной записи, внедрение уведомлений по электронной почте, расширение функциональности формирования приказов, переход на промышленную СУБД, предоставление заведующему кафедрой возможности настройки лимита заявок, реализация модуля планирования консультаций, а также поддержка формата «Стартап как диплом» и адаптация системы под межкафедральное руководство.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Приказ Минобрнауки России от 29.06.2015 № 636 (ред. от 11.03.2022) «Об утверждении Порядка проведения государственной итоговой аттестации по образовательным программам высшего образования – программам бакалавриата, программам специалитета и программам магистратуры». – Текст : непосредственный.
2. Черных Д.В., Фурер О.В. Цифровая трансформация высшего образования // Мир педагогики и психологии. – 2025. – № 1 (102). – URL: <https://scipress.ru/pedagogy/articles/tsifrovaya-transformatsiya-vysshego-obrazovaniya.html> (дата обращения: 28.02.2026). – Текст : электронный.
3. Калугина Т.Н., Чибисгулева Т.И. Анализ процессов цифровой трансформации образовательной среды на примере российского вуза // Научное мнение. – 2024. – № 3. – С. 71–78. – Текст : непосредственный.
4. Рабаданов М.Х., Магомедова С.Р., Ниналалов С.А. Некоторые аспекты цифровой трансформации университета // Вестник кибернетики. – 2025. – Т. 24, № 2. – С. 58–65. – Текст : непосредственный.
5. Веселицкий О.И. Роль вузов в реализации программы «Цифровая экономика» // Новые информационные технологии в образовании. – 2018. – Ч. 2. – С. 248–252. – Текст : непосредственный.
6. Московское агентство новостей. Около 823 тыс. человек окончили российские вузы в этом году. – 2024. – URL: <https://www.mskagency.ru/materials/3430001> (дата обращения: 08.06.2026). – Текст : электронный.
7. Объясняем.рф. В этом году вузы выпустят около 824 тысяч студентов. – 2024. – URL: <https://объясняем.рф/articles/news/v-etom-godu-vuzy-vypustyat-okolo-824-tysyach-studentov/> (дата обращения: 08.06.2026). – Текст : электронный.
8. Московский политехнический университет. Требования и методические рекомендации для выпускных квалификационных работ по направлению

09.03.03 «Прикладная информатика» (образовательная программа «Информационные технологии управления бизнесом»). – Москва, 2026. – 23 с. – Текст : непосредственный.

9. Федеральный государственный образовательный стандарт высшего образования – бакалавриат по направлению подготовки 09.03.03 «Прикладная информатика» (утв. Приказом Минобрнауки России от 19.09.2017 № 922). – Текст : непосредственный.

10. Родюков А.В., Сосенушкин С.Е., Харин А.А. Перспективы автоматизации управления деятельностью образовательных организаций // Новые информационные технологии в образовании. – 2018. – Ч. 2. – С. 20–24. – Текст : непосредственный.

11. Исаева Е.С. Современные LMS платформы дистанционного обучения: анализ и сравнение // Педагогика. Вопросы теории и практики. – 2021. – Т. 6, № 6. – С. 1045–1050. – Текст : непосредственный.

12. Белозёрова С.И., Чуйко О.И. Опыт применения LMS Moodle для создания и сопровождения учебных курсов // Современные проблемы науки и образования. – 2019. – № 6. – URL: <https://science-education.ru/ru/article/view?id=29382> (дата обращения: 28.02.2026). – Текст : электронный.

13. Doran G.T. There's a S.M.A.R.T. Way to Write Management's Goals and Objectives // Management Review. – 1981. – Vol. 70, No. 11. – P. 35–36. – Текст : непосредственный.

14. Turner J.R. The Handbook of Project-Based Management: Leading Strategic Change in Organizations. – 3rd ed. – New York : McGraw-Hill, 2009. – 412 p. – Текст : непосредственный.

15. ISO 21500:2021. Project management — A guide to project management. – Geneva : ISO, 2021. – 68 p. – Текст : непосредственный.

16. Буч Г., Рамбо Д., Якобсон А. Язык UML. Руководство пользователя. – 2-е изд. – Москва : ДМК Пресс, 2021. – 496 с. – Текст : непосредственный.

17. Ларман К. Применение UML и шаблонов проектирования. – 3-е изд. – Москва : Вильямс, 2020. – 736 с. – Текст : непосредственный.
18. EleWise. Графический язык моделирования бизнес-процессов BPMN. Спецификация (избранные главы). – Москва : EleWise, 2009. – 120 с. – Текст : непосредственный.
19. Федоров И.Г. Нотация BPMN 2.0. Стандарт ISO/IEC 19510:2013 для создания исполняемых моделей бизнес-процессов. – Москва : Российский экономический университет имени Г. В. Плеханова, 2022. – 180 с. – Текст : непосредственный.
20. Сафиуллин М.Р., Смольникова К.Р. Моделирование бизнес-процессов в эргатических системах // Теоретическая и прикладная экономика. – 2026. – № 2. – С. 84–105. – DOI: 10.25136/2409-8647.2026.2.79408. – Текст : непосредственный.
21. Поступаева С.Г. Построение функциональных диаграмм в нотации IDEF0: лабораторный практикум. – Волгоград : ВолгГТУ, 2022. – 96 с. – Текст : непосредственный.
22. Олифер В.Г., Олифер Н.А. Компьютерные сети. Принципы, технологии, протоколы. – Санкт-Петербург : Питер, 2023. – 1008 с. – Текст : непосредственный.
23. Django Software Foundation. Django Documentation (Version 4.2). – URL: <https://docs.djangoproject.com/en/4.2/> (дата обращения: 25.05.2026). – Текст : электронный.
24. Дронов В.А. Django: практика создания веб-сайтов на Python. – 2-е изд. – Санкт-Петербург : БХВ-Петербург, 2022. – 592 с. – Текст : непосредственный.
25. Меле А. Django 4 в примерах / пер. с англ. – Москва : ДМК Пресс, 2023. – 800 с. – Текст : непосредственный.
26. Лутц М. Изучаем Python. – 5-е изд. – Санкт-Петербург : Питер, 2022. – 1312 с. – Текст : непосредственный.

27. Порфирьев А. Встроенные механизмы безопасности фреймворков Python // УЦСБ: Центр информационной безопасности. – 2024. – URL: <https://www.ussc.ru/center/news/vstroennye-mekhanizmy-bezopasnosti-freymvorkov-python/> (дата обращения: 28.02.2026). – Текст : электронный.

28. Полат Е.С., Бухаркина М.Ю. Новые педагогические и информационные технологии в системе образования. – Москва : Академия, 2010. – 368 с. – Текст : непосредственный.

29. Попова Н.А., Егорова Е.С. Основные направления интеллектуального анализа данных в сфере образования // Известия КБНЦ РАН. – 2024. – Т. 26, № 5. – С. 94–106. – Текст : непосредственный.

30. SQLite Consortium. SQLite Documentation. – URL: <https://www.sqlite.org/docs.html> (дата обращения: 28.02.2026). – Текст : электронный.

31. PostgreSQL Global Development Group. PostgreSQL 16 Documentation. – URL: <https://www.postgresql.org/docs/16/> (дата обращения: 28.02.2026). – Текст : электронный.

32. HTML Living Standard – WHATWG. – URL: <https://html.spec.whatwg.org/> (дата обращения: 28.02.2026). – Текст : электронный.

33. Bootstrap Team. Bootstrap 5.3 Documentation. – URL: <https://getbootstrap.com/docs/5.3> (дата обращения: 25.05.2026). – Текст : электронный.

34. Pandas documentation. – URL: <https://pandas.pydata.org/docs/> (дата обращения: 25.05.2026). – Текст : электронный.

35. Python-docx documentation. – URL: <https://python-docx.readthedocs.io/en/latest/> (дата обращения: 25.05.2026). – Текст : электронный.

36. Дейт К.Дж. Введение в системы баз данных. – 8-е изд. – Москва : Вильямс, 2006. – 1328 с. – Текст : непосредственный.

37. Швецов В.И. Базы данных: учебное пособие. – Москва : Вузовский учебник, 2019. – 368 с. – Текст : непосредственный.
38. Dietrich S.W., Urban S.D. An Advanced Course in Database Systems: Beyond Relational Databases. – Pearson, 2005. – 302 p. – ISBN 9780130428981. – Текст : непосредственный.
39. Barabási A.-L. Network Science. – Cambridge : Cambridge University Press, 2016. – 474 p. – Текст : непосредственный.
40. Petrosian A., Kazarian A. Identification and Prioritization of Risks in the Graduate Thesis Preparation Process Based on a Cognitive Model Formed from Psycholinguistic Experiment Data // 2026 International Russian Smart Industry Conference (SmartIndustryCon). – Sochi, Russian Federation, 2026. – P. 1062–1068. – DOI: 10.1109/SmartIndustryCon68821.2026.11492985. – Текст : непосредственный.
41. ГОСТ 19.701-90 (ИСО 5807-85). Единая система программной документации. Схемы алгоритмов, программ, данных и систем. – Москва : Стандартинформ, 1990. – 23 с. – Текст : непосредственный.
42. ГОСТ 34.601-90. Информационная технология. Комплекс стандартов на автоматизированные системы. Автоматизированные системы. Стадии создания. – Москва : Стандартинформ, 1992. – 15 с. – Текст : непосредственный.
43. ISO/IEC 25010:2023. Systems and software engineering – Systems and software Quality Requirements and Evaluation (SQuaRE) — System and software quality models. – Geneva : ISO, 2023. – 56 p. – Текст : непосредственный.
44. ГОСТ Р ИСО 9241-210-2016. Эргономика взаимодействия человек-система. Человеко-ориентированное проектирование. – Москва : Стандартинформ, 2016. – 28 с. – Текст : непосредственный.
45. Django Software Foundation. Writing your first Django app, part 1 – Project structure. – URL: <https://docs.djangoproject.com/en/4.2/intro/tutorial01/> (дата обращения: 25.05.2026). – Текст : электронный.

46. Django Software Foundation. Customizing authentication in Django – AbstractUser. – URL: <https://docs.djangoproject.com/en/4.2/topics/auth/customizing/#substituting-a-custom-user-model> (дата обращения: 25.05.2026). – Текст : электронный.

47. Django Software Foundation. Signals in Django. – URL: <https://docs.djangoproject.com/en/4.2/topics/signals/> (дата обращения: 25.05.2026). – Текст : электронный.

48. Django Software Foundation. The Django admin site. – URL: <https://docs.djangoproject.com/en/4.2/ref/contrib/admin/> (дата обращения: 25.05.2026). – Текст : электронный.

49. Django Software Foundation. ModelAdmin — customization of admin interface. – URL: <https://docs.djangoproject.com/en/4.2/ref/contrib/admin/#modeladmin-options> (дата обращения: 25.05.2026). – Текст : электронный.

50. Django Software Foundation. The Django template language — for designers. – URL: <https://docs.djangoproject.com/en/4.2/ref/templates/language/> (дата обращения: 25.05.2026). – Текст : электронный.

51. Django Software Foundation. Cross Site Request Forgery protection. – URL: <https://docs.djangoproject.com/en/4.2/ref/csrf/> (дата обращения: 25.05.2026). – Текст : электронный.

52. Куликов С.С. Тестирование программного обеспечения. Базовый курс. – 3-е изд., версия 3.2.6. – EPAM Systems, 2024. – 301 с. – URL: https://svyatoslav.biz/software_testing_book/ (дата обращения: 28.02.2026). – Текст : электронный.

53. IEEE 829-2008. Standard for Software and System Test Documentation. – New York : IEEE, 2008. – 44 p. – Текст : непосредственный.

54. ГОСТ Р ИСО/МЭК 12207-2010. Информационная технология. Процессы жизненного цикла программных средств. – Москва : Стандартинформ, 2011. – 68 с. – Текст : непосредственный.

55. ISO/IEC 12207:2017. Systems and software engineering – Software life cycle processes. – Geneva : ISO, 2017. – 120 p. – Текст : непосредственный.

56. Ollama. Qwen2.5:7b-instruct : библиотека моделей. – URL: <https://ollama.com/library/qwen2.5:7b-instruct> (дата обращения: 08.06.2026). – Текст : электронный.

57. Галынчик Т.А., Данилова С.В., Агеев Я.Д. Реализация программы «Стартап как диплом» в системе высшего образования // Научно-методический электронный журнал «Концепт». – 2025. – № 5 (май). – С. 352–368. – URL: <http://e-koncept.ru/2025/251099.htm> (дата обращения: 28.02.2026). – Текст : электронный.

58. Московский политехнический университет. Основная профессиональная образовательная программа высшего образования – программа бакалавриата по направлению подготовки 09.03.03 «Прикладная информатика» (профиль «Информационные технологии управления бизнесом»). – Москва, 2023. – 180 с. – Текст : непосредственный.

59. Вердовский О.В., Шевчук Е.В., Шпак А.В. Автоматизация процесса размещения выпускных квалификационных работ обучающихся в электронную библиотеку // Интерэкспо ГЕО-Сибирь. – 2023. – Т. 1, № 1. – С. 252–257. – Текст : непосредственный.

ПРИЛОЖЕНИЕ А

Рисунки

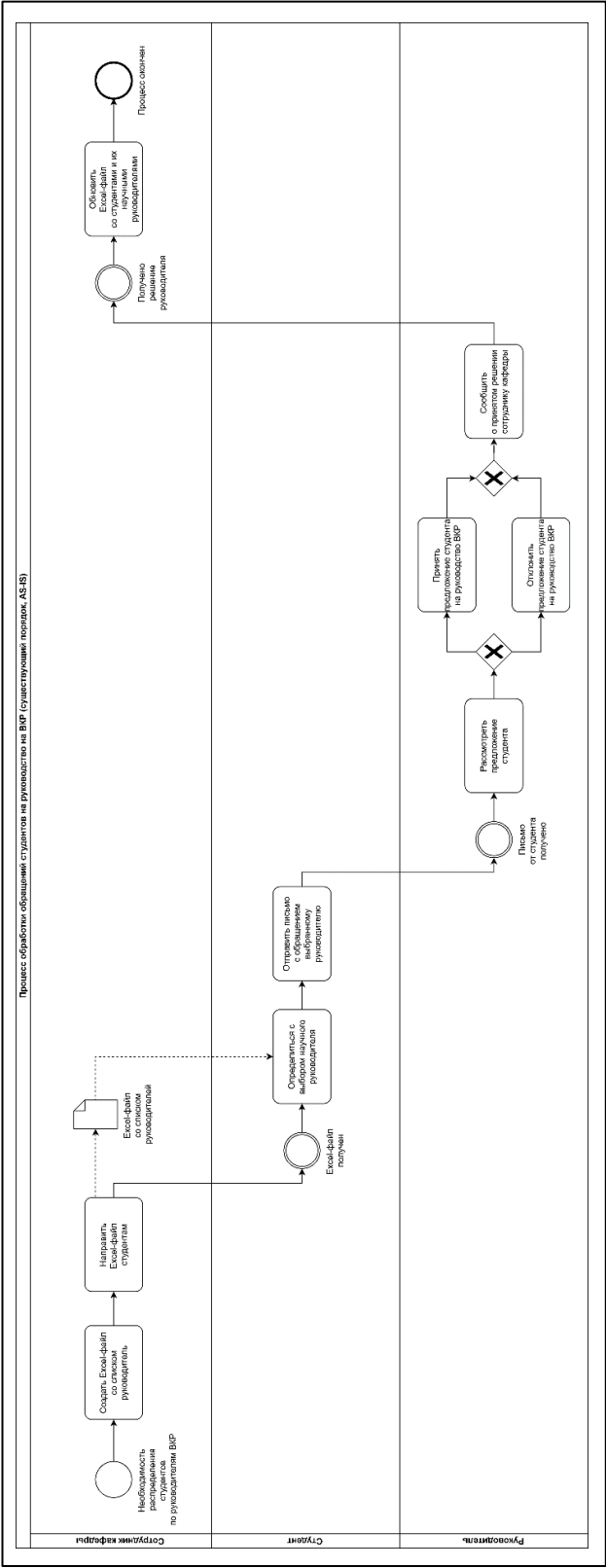


Рисунок А.1 – Диаграмма бизнес-процесса «Управление заявками на руководство ВКР» в нотации BPMN 2.0 (модель AS-IS)

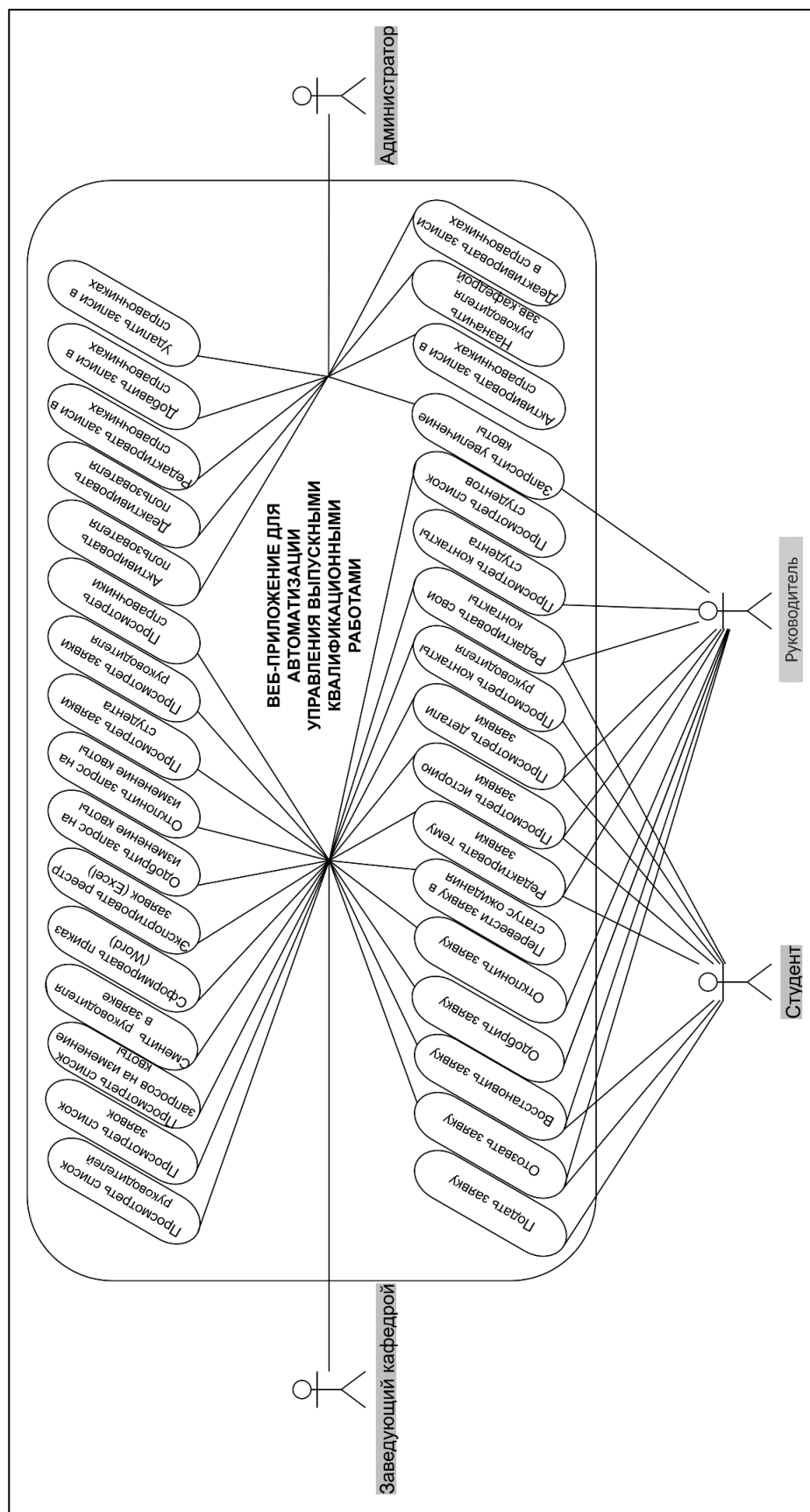


Рисунок А.2 – диаграмма прецедентов UML

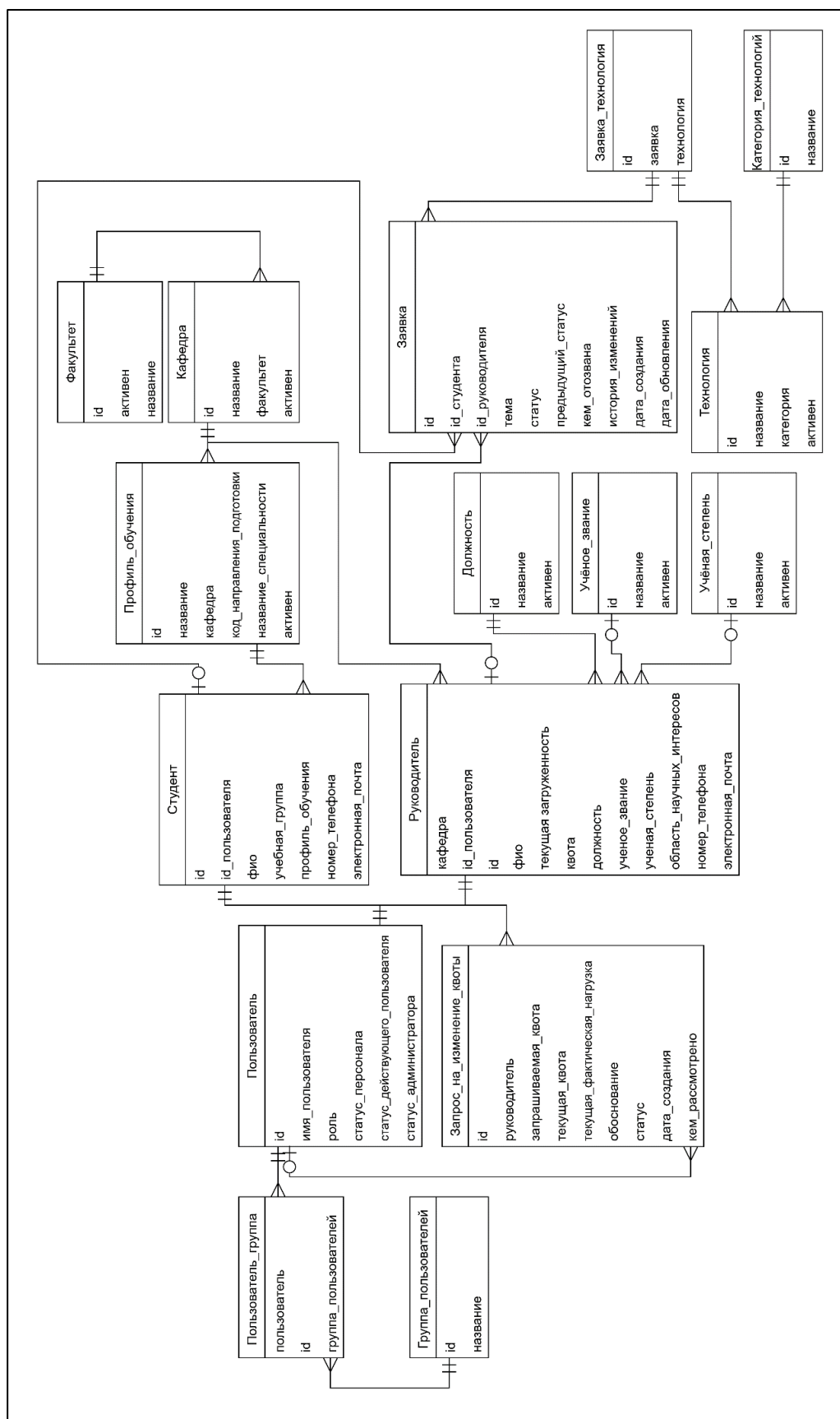


Рисунок А.3 – Инфологическая модель базы данных в нотации Мартина

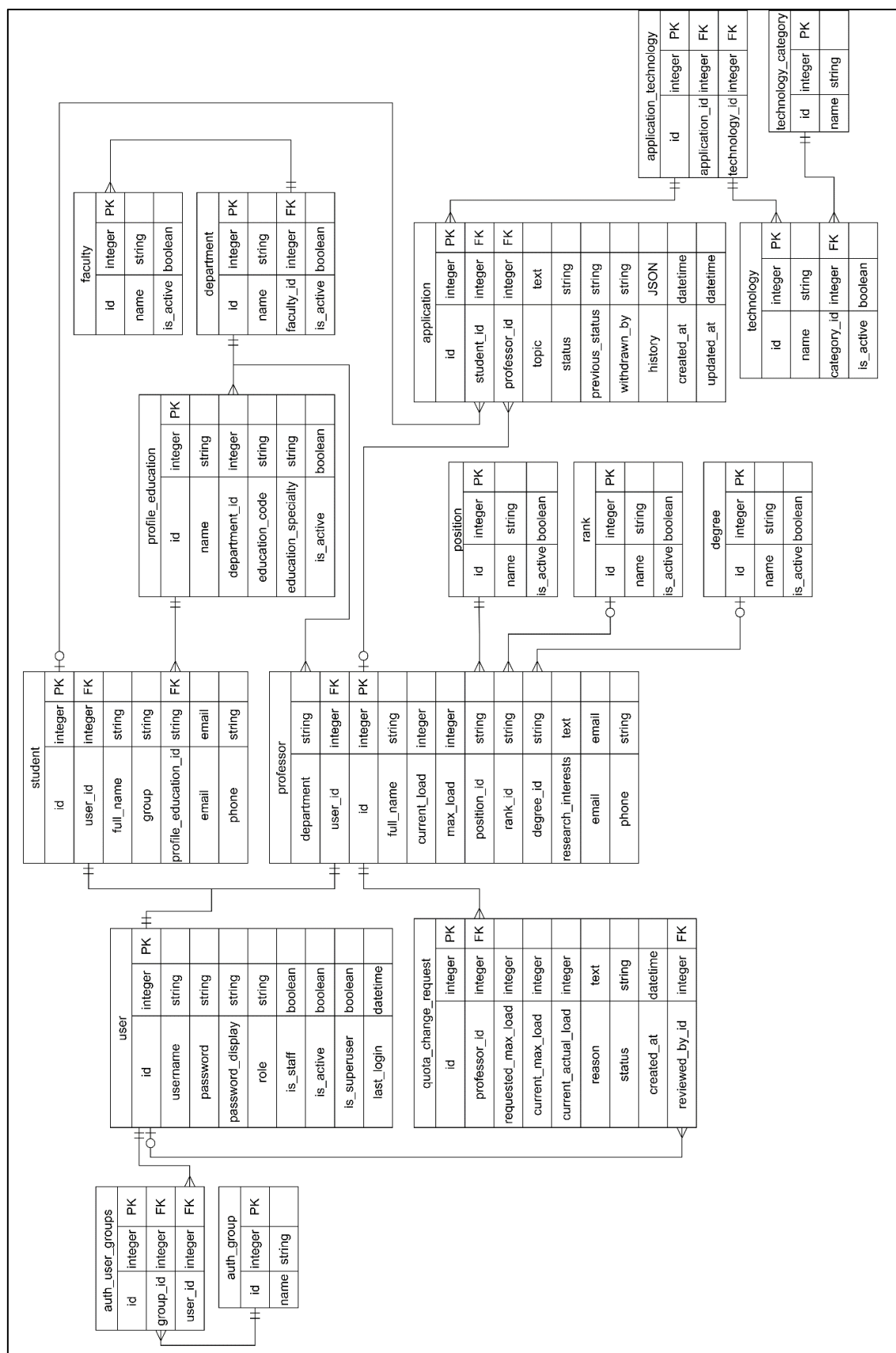


Рисунок А.4 – Даталогическая модель базы данных в нотации Мартина

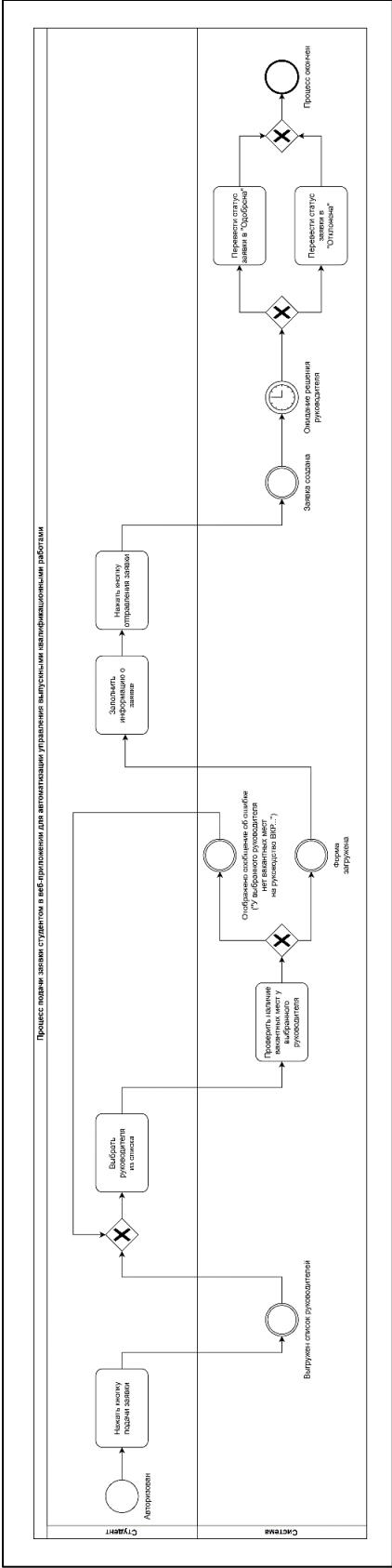


Рисунок А.5 – Диаграмма процесса «Подача заявки в системе студентом» в нотации BPMN 2.0 (TO-BE)

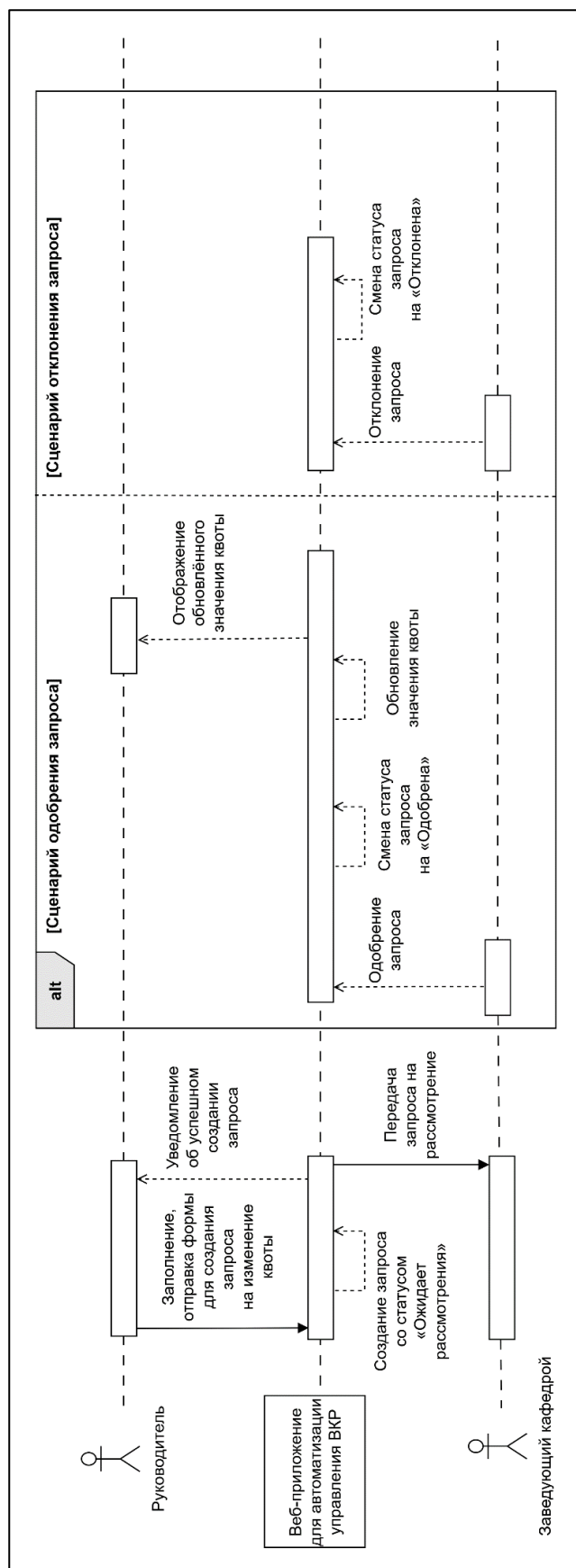


Рисунок А.6 – Диаграмма последовательностей (UML) процесса «Запрос руководителя на изменение КВОТЫ»

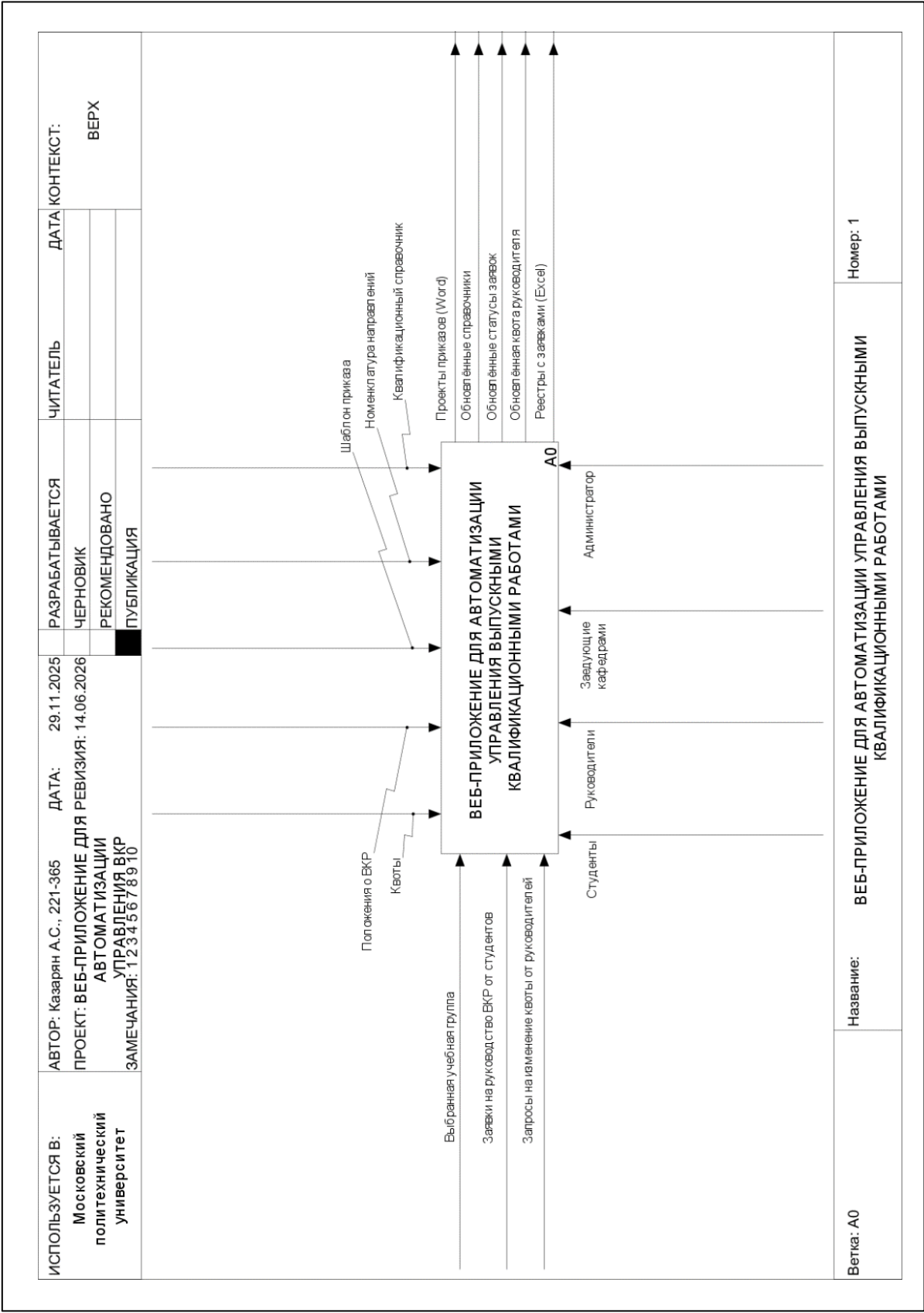


Рисунок А.7 – Контекстная диаграмма IDEF0 (A-0) процесса управления ВКР (ТО-ВЕ)

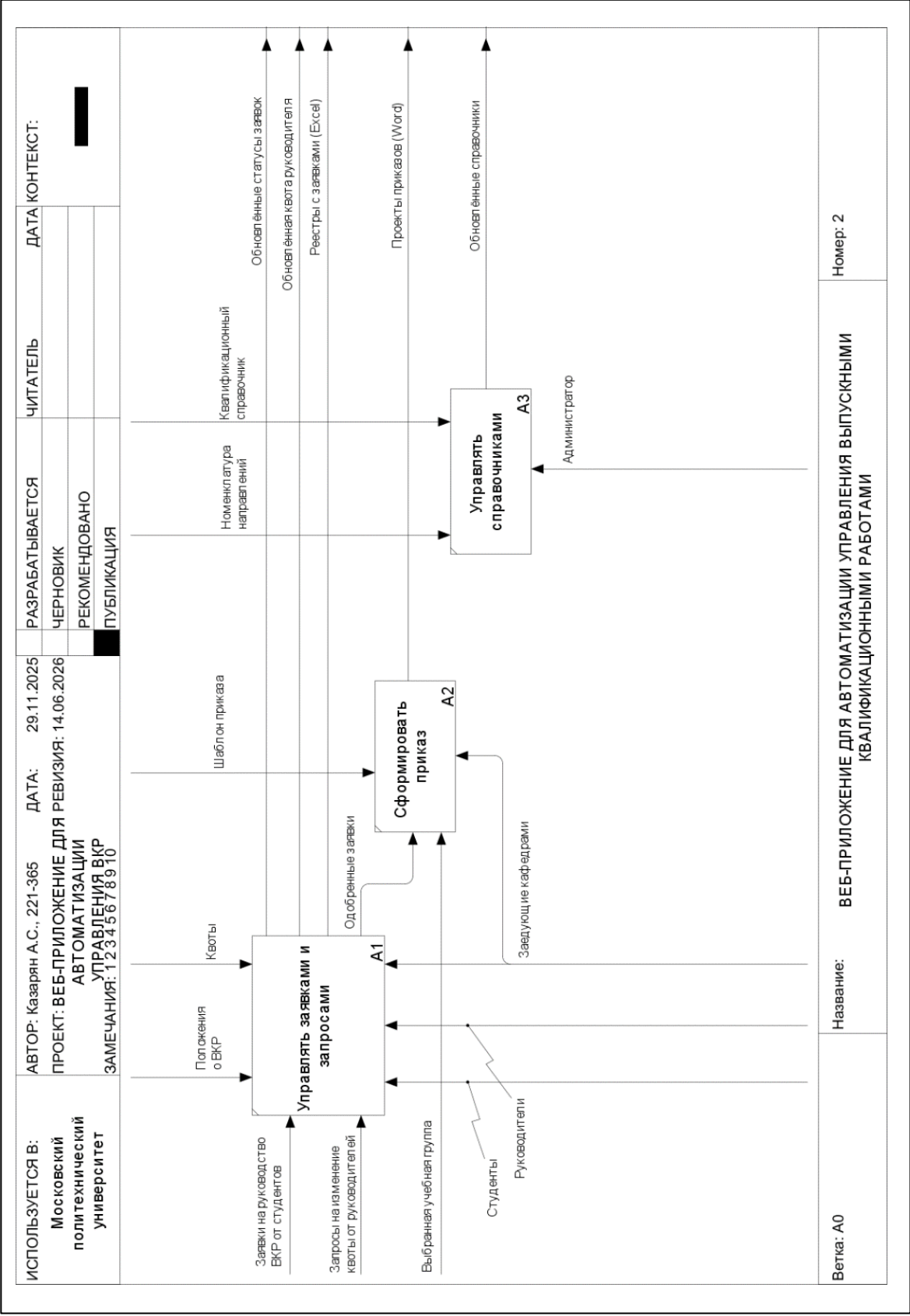


Рисунок А.8 – Диаграмма декомпозиции IDEF0 (A0) процесса управления ВКР (ТО-ВЕ)

Соколов Илья Владимирович (Студент)

Выйти

Веб-приложение для автоматизации управления ВКР

Найдено преподавателей: 8

Список преподавателей

Все преподаватели					
№	ФИО преподавателя	Кафедра	Область научных интересов	Действия	Загруженность
1	Волкова Наталья Сергеевна	Информационная безопасность (Информационных технологий)	Бизнес-аналитика, BI-системы, Data Science	Подать заявку	7/11
2	Кузнецов Александр Алексеевич	Информационная безопасность (Информационных технологий)	Облачные технологии, DevOps, контейнеризация	Подать заявку	4/10
3	Петров Сергей Михайлович	Информационная безопасность (Информационных технологий)	Базы данных, SQL, NoSQL, оптимизация запросов	Подать заявку	3/12
4	Попов Александр Ильич	Информационная безопасность (Информационных технологий)	Естественно-языковая обработка текстов (NLP)	Подать заявку	6/10
5	Смирнова Ольга Андреевна	Информационная безопасность (Информационных технологий)	Бизнес-аналитика, BI-системы, Data Science	Подать заявку	6/10
6	Соколова Светлана Алексеевна	Информационная безопасность (Информационных технологий)	Естественно-языковая обработка текстов (NLP)	Подать заявку	6/6

Рисунок А.9 – Личный кабинет студента, список руководителей

Подача заявки преподавателю: Волкова Наталья Сергеевна

Тема дипломной работы *

Введите тему вашей дипломной работы

Описание работы

Опишите подробнее, что вы планируете сделать...

Необязательно, но поможет преподавателю лучше понять Вашу идею

Технологии

Веб-фреймворки и анализ данных

☐ Docker

☐ FastAPI

☐ Pandas

☐ PostgreSQL

☐ React

☐ Scikit-learn

Корпоративная разработка

☐ Git

☐ Hibernate

☐ Java

☐ Maven

☐ MySQL

☐ Spring Boot

Сетевые технологии и безопасность

☐ Bash

☐ Metasploit

☐ Nmap

☐ OpenSSL

☐ Python

☐ Wireshark

Другие технологии

Технологии, которых нет выше

Отправить заявку

Отмена

Рисунок А.10 – Форма подачи заявки



Рисунок А.11 – Личный кабинет студента, список поданных заявок

Соколова Светлана Алексеевна (Преподаватель)

Выйти

Веб-приложение для автоматизации управления ВКР

Контакты: sokolova.s.a@mospolyspu.ru , +79844232119

Научные интересы: Естественно-языковая обработка текстов (NLP)

Загруженность: 5/6 студентов

Ожидают (0)

Одобрены (5)

Отклонены (0)

Отозваны (2)

Все

Входящие заявки

Все заявки

№	Студент	Профиль	Группа	Тема работы	Действия	Статус	Действия
1	Соколов Илья Владимирович	Информационная безопасность (Информационная безопасность)	221-109	Исследование когнитивных наук с помощью современных методов машинного обучения в условиях цифровизации обучения		Одобрена	Отозвать
2	Новиков Артём Алексеевич	Информационная безопасность (Информационная безопасность)	221-108	Автоматизация документооборота		Отозвана	Восстановить
3	Новикова Елена Михайловна	Информационная безопасность (Информационная)	221-108	Исследование методов машинного обучения		Одобрена	Отозвать

? Помощь

Рисунок А.12 – Личный кабинет руководителей

Запрос на изменение квоты

Текущая квота: 5

Запрашиваемая квота:

5

Обоснование (необязательно):

Укажите причину изменения квоты...

Внимание! Ваш запрос будет направлен заведующему кафедры на рассмотрение. Текущая квота изменится только после одобрения.

Отмена

Отправить запрос

Рисунок А.13 – Форма запроса изменения квоты

История заявки от 07.05.2026 19:37

Экспорт

Орлова Ирина Владимировна
07.05.2026 19:37

Заявка подана студентом

Тема: Исследование методов машинного обучения

Максимова Анна Никитична
07.05.2026 20:39

Заявка одобрена преподавателем

Заявка одобрена

Орлова Ирина Владимировна
11.05.2026 02:24

Тема изменена студентом

Новая тема: Исследование методов машинного обучения в разработке прототипов производственных деталей

Максимова Анна Никитична
11.05.2026 12:27

Тема изменена преподавателем

Новая тема: Разработка веб-сервиса для автоматизации планирования маршрутов технического обслуживания парка городского пассажирского транспорта

Заккрыть

Рисунок А.14 – История заявки

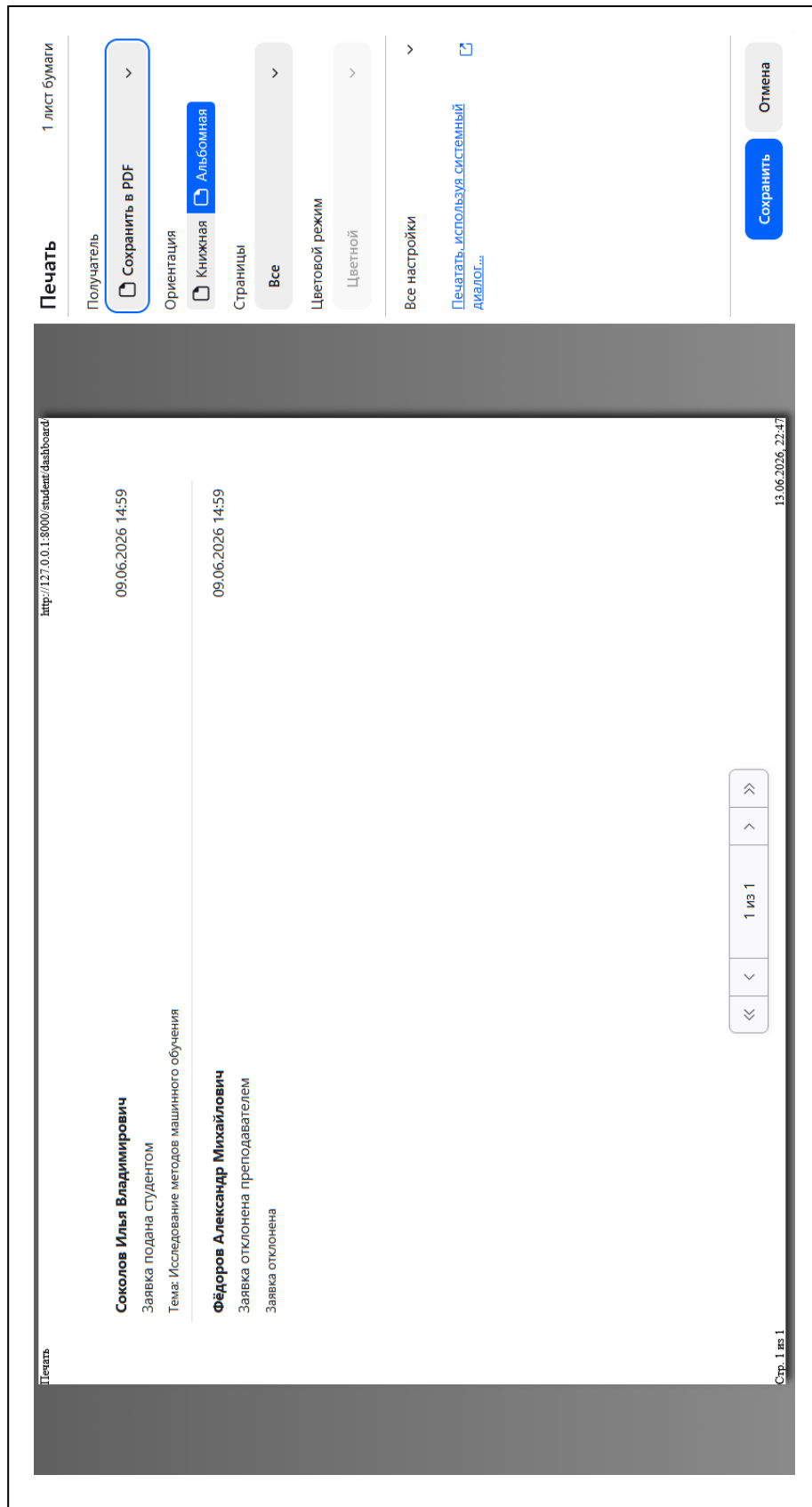


Рисунок А.15 – Экспорт истории заявки с помощью штатных средств браузера



Рисунок А.16 – Административная панель заведующего кафедрой

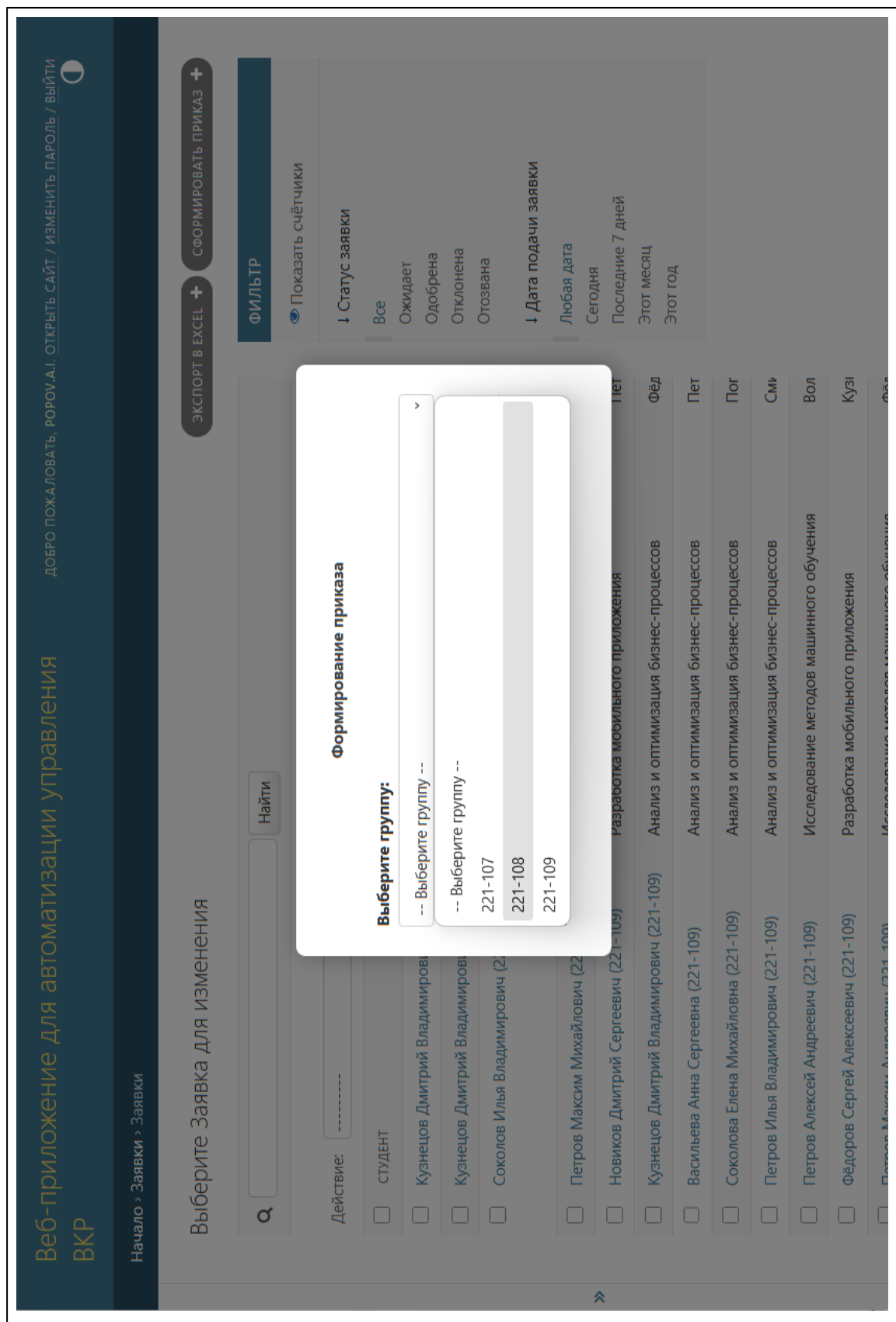


Рисунок А.17 – Выбор учебной группы для формирования проекта приказа

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
«МОСКОВСКИЙ ПОЛИТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»

(МОСКОВСКИЙ ПОЛИТЕХ)

ПРИКАЗ

10.05.2026

№ _____ -

О назначении руководителей
и утверждении тем выпускных
квалификационных работ

В соответствии с федеральным государственным образовательным стандартом высшего образования по направлению подготовки 38.03.01 «Экономика»

ПРИКАЗЫВАЮ:

1. Назначить руководителей и утвердить темы выпускных квалификационных работ по кафедре «Экономика и организация» факультета Экономики и управления следующим обучающимся:

учебная группа 221-108:

№ п/п	Фамилия Имя Отчество	Тема выпускной квалификационной работы	Руководитель (должность, ФИО)
1.	Алексеева Дарья Ильинична	Разработка веб-приложения на Java	Ассистент Соколов Михаил Михайлович
2.	Васильев Артём Егорович	Автоматизация тестирования	Ассистент Соколов Михаил Михайлович
3.	Васильев Михаил Ильич	Разработка мобильного приложения на JavaScript	Ассистент Соколова Мария Дмитриевна
4.	Васильева Галина Алексеевна	Исследование методов анализа данных	Преподаватель Смирнов Владислав Александрович
5.	Иванова Ирина Сергеевна	Разработка API на Java	Преподаватель Кузнецов Павел Андреевич
6.	Козлов Илья Максимович	Проектирование базы данных	Ассистент Соколов Михаил Михайлович
7.	Козлова Ольга Михайловна	Автоматизация тестирования	Старший преподаватель Иванов Дмитрий Никитич

О назначении руководителей и утверждении тем выпускных квалификационных работ – 09-21
Центр по работе со студентами, отделение «На Большой Семеновской» (очная форма обучения) Факультет Экономики и управления
Исп.: Фамилия Имя Отчество; тел.: +7 () - - , дата формирования: 10.05.2026
ИД

Рисунок А.18 – Первая страница сформированного проекта приказа

8.	Макаров Кирилл Максимович	Разработка мобильного приложения на Java	Доцент Морозова Галина Павловна
9.	Никитина Светлана Александровна	Проектирование базы данных	Доцент Михайлова Елена Павловна
10.	Орлов Кирилл Павлович	Разработка микросервиса	Доцент Никитин Дмитрий Романович
11.	Павлова Юлия Максимовна	Исследование методов оптимизации	Доцент Никитин Дмитрий Романович
12.	Соколова Людмила Артёмовна	Проектирование базы данных	Старший преподаватель Николаева Анна Егоровна

Основание: протокол заседания кафедры «Экономика и организация» от _____. № ____/____-_____

2. Контроль за исполнением приказа возложить на соответствующего заведующего кафедрой и/или руководителя образовательной программы.

Декан факультета

О назначении руководителей и утверждении тем выпускных квалификационных работ – 09-21
 Центр по работе со студентами, отделение «На Большой Семеновской» (очная форма обучения) Факультет Экономики и управления
 Исп.: Фамилия Имя Отчество; тел.: +7 (____) ____-____, дата формирования: 10.05.2026
 ИД

Рисунок А.19 – Вторая страница сформированного проекта приказа

	A		B	C		D	E	F	G	H
	ФИО студента	Группа	Тема ВКР	Научный руководитель	Кафедра	Статус	Дата создания	Дата обновления		
1	Алексеев Тимур Максимович	221-107	Разработка чат- бота	Никитин Дмитрий Романович	Менеджмент	Одобрена	07.05.2026	07.05.2026		
2	Алексеева Дарья Ильинична	221-108	Разработка веб-приложения на Java	Соколов Михаил Михайлович	Менеджмент	Одобрена	07.05.2026	07.05.2026		
3	Васильев Артём Егорович	221-108	Автоматизация тестирования	Соколов Михаил Михайлович	Менеджмент	Одобрена	07.05.2026	07.05.2026		
4	Васильев Михаил Ильич	221-108	Автоматизация бизнес-процессов	Никитин Дмитрий Романович	Менеджмент	Одобрена	07.05.2026	07.05.2026		
5	Васильев Михаил Ильич	221-108	Разработка мобильного приложения на JavaScript	Соколова Мария Дмитриевна	Менеджмент	Одобрена	07.05.2026	07.05.2026		
6	Васильева Галина Алексеевна	221-108	Автоматизация бизнес-процессов	Соколова Мария Дмитриевна	Менеджмент	Одобрена	07.05.2026	07.05.2026		
7	Волкова Юлия Романовна	221-109	Проектирование архитектуры ПО	Васильев Тимур Кириллович	Менеджмент	Одобрена	07.05.2026	07.05.2026		
8	Волкова Юлия Романовна	221-109	Исследование методов машинного обучения	Кузнецова Екатерина Александровна	Менеджмент	Одобрена	07.05.2026	07.05.2026		
9	Егорова Галина Михайловна	221-108	Разработка веб-сервиса	Соколов Михаил Михайлович	Менеджмент	Отклонена	07.05.2026	07.05.2026		
10	Иванов Кирилл Дмитриевич	221-107	Исследование методов анализа данных	Никитин Дмитрий Романович	Менеджмент	Одобрена	07.05.2026	07.05.2026		
11	Иванов Кирилл Дмитриевич	221-107	Исследование методов анализа данных	Никитин Дмитрий Романович	Менеджмент	Одобрена	07.05.2026	07.05.2026		
12	Иванова Елена Егоровна	221-107	Автоматизация тестирования	Никитин Дмитрий Романович	Менеджмент	Одобрена	07.05.2026	07.05.2026		
13	Иванова Елена Сергеевна	221-108	Разработка API на Python	Никитин Дмитрий Романович	Менеджмент	Одобрена	07.05.2026	07.05.2026		
14	Иванова Ирина Сергеевна	221-108	Автоматизация тестирования	Васильев Тимур Кириллович	Менеджмент	Одобрена	07.05.2026	07.05.2026		
15	Козлов Илья Максимович	221-108	Проектирование базы данных	Соколов Михаил Михайлович	Менеджмент	Одобрена	07.05.2026	07.05.2026		
16	Козлов Максим Максимович	221-109	Разработка веб-приложения на Python	Васильев Тимур Кириллович	Менеджмент	Ожидает рассмотрения	07.05.2026	07.05.2026		
17	Кузнецов Дмитрий Павлович	221-109	Разработка веб-сервиса	Васильев Тимур Кириллович	Менеджмент	Ожидает рассмотрения	07.05.2026	07.05.2026		
18	Лебедева Ирина Ильинична	221-107	Проектирование архитектуры ПО	Никитин Дмитрий Романович	Менеджмент	Ожидает рассмотрения	07.05.2026	07.05.2026		
19	Михайлова Надежда Михайловна	221-109	Разработка мобильного приложения на JavaScript	Никитин Дмитрий Романович	Менеджмент	Одобрена	07.05.2026	07.05.2026		
20	Морозов Сергей Павлович	221-109	Разработка веб-сервиса	Васильев Тимур Кириллович	Менеджмент	Одобрена	07.05.2026	07.05.2026		
21	Морозов Сергей Павлович	221-109	Разработка веб-сервиса	Соколов Михаил Михайлович	Менеджмент	Одобрена	07.05.2026	07.05.2026		
22	Морозова Елена Павловна	221-109	Исследование методов анализа данных	Васильев Тимур Кириллович	Менеджмент	Отклонена	07.05.2026	07.05.2026		
23	Никитин Егор Сергеевич	221-109	Разработка веб-приложения на JavaScript	Никитин Дмитрий Романович	Менеджмент	Одобрена	07.05.2026	07.05.2026		
24	Никитин Роман Андреевич	221-107	Автоматизация бизнес-процессов	Соколова Мария Дмитриевна	Менеджмент	Ожидает рассмотрения	07.05.2026	07.05.2026		
25	Николаев Дмитрий Максимович	221-108	Проектирование базы данных	Васильев Тимур Кириллович	Менеджмент	Ожидает рассмотрения	07.05.2026	07.05.2026		
26	Николаев Егор Максимович	221-109	Автоматизация документооборота	Васильев Тимур Кириллович	Менеджмент	Одобрена	07.05.2026	07.05.2026		
27	Николаев Павел Дмитриевич	221-107	Исследование методов анализа данных	Соколов Михаил Михайлович	Менеджмент	Одобрена	07.05.2026	07.05.2026		
28	Новиков Тимур Максимович	221-109	Автоматизация тестирования	Кузнецова Екатерина Александровна	Менеджмент	Отклонена	07.05.2026	07.05.2026		

Рисунок А.20 – Экспортированный реестр заявок

ПРИЛОЖЕНИЕ Б

Таблицы

Таблица Б.1 – Классификация требований к системе

Уровень	Тип	Требование	Источник требования
1	2	3	4
Бизнес-требования	Функциональное	Система должна автоматизировать процесс выбора тем и научных руководителей по ВКР	Анализ процессов AS-IS
Пользовательские требования	Функциональное	Студент должен иметь возможность подать заявку выбранному руководителю с указанием темы работы	Анализ пользовательских сценариев
		Руководитель должен видеть список входящих заявок и иметь возможность их одобрить или отклонить	Наблюдение за рабочим процессом
	Нефункциональное	Интерфейс должен быть интуитивно понятен для пользователей со средним уровнем компьютерной грамотности	Анализ целевой аудитории
Системные требования	Функциональное	При одобрении заявки система должна автоматически уменьшать количество вакантных мест у соответствующего руководителя на единицу	Вывод из бизнес-правил контроля количества занятых мест
		Система должна предотвращать создание дублирующих заявок от одного студента к одному руководителю	Анализ проблемы дублирования и неопределенности в текущем процессе
		Система должна вести журнал изменений для каждой заявки в формате JSON	Потребность в аудите действий участников процесса
		Система должна генерировать проект приказа о назначении руководителей в формате Word с оформлением по шаблону университета	Анализ документооборота
		Система должна предоставлять руководителю возможность отправлять запрос на изменение квоты на руководство ВКР с указанием обоснования, а заведующему кафедрой — возможность одобрить или отклонить такой запрос	Анализ процесса управления нагрузкой

Продолжение таблицы Б.1

1	2	3	4
		Система должна позволять студенту и руководителю отозвать заявку на любом этапе. При отзыве одобренной заявки количество вакантных мест у руководителя должно автоматически увеличиваться. Система также должна предоставлять возможность восстановления отозванной заявки с возвратом к предыдущему статусу и соответствующим обновлением количества вакантных мест у руководителя	Анализ пользовательских сценариев
		Система должна ограничивать количество активных заявок от одного студента (не более трёх). Активными считаются заявки со статусами «Ожидает» и «Одобрена». Отклонённые и отозванные заявки не учитываются. При попытке подать заявку сверх лимита система должна блокировать её создание и выводить сообщение об ошибке.	Анализ целевой аудитории и требований к предотвращению нецелевого использования системы
	Нефункциональное	Система должна поддерживать интеграцию с существующей системой единого входа (SSO) университета для аутентификации пользователей без необходимости хранения их паролей в промышленной эксплуатации.	Архитектурное решение (требование к промышленной эксплуатации)
		Система должна поддерживать русскоязычный интерфейс	Анализ целевой аудитории
		Система должна поддерживать интерфейс для пользователей с ограничениями зрения	Анализ целевой аудитории
		Система должна поддерживать одновременную работу не менее 100 пользователей при времени отклика на стандартные операции не более 2 секунд	Расчет пиковой нагрузки
		Система должна быть доступна для авторизованных пользователей 24/7 в период подачи заявок	Требование непрерывности процесса

Таблица Б.2 – Матрица ролей и прав доступа

Действие	Студент	Руководитель	Заведующий кафедрой	Администратор
Просмотр списка руководителей	да, своей кафедры	нет	да, своей кафедры	да, всего университета
Просмотр списка студентов	нет	нет	да, своей кафедры	да, всего университета
Просмотр списка заявок	да, исходящих	да, входящих	да, своей кафедры	да, всего университета
Подача заявки	да	нет	нет	нет
Просмотр информации о руководителе	да	нет	да	да
Просмотр информации о студенте	нет	да	да	да
Запрос на изменение квоты	нет	да	нет	нет
Одобрение/отклонение заявки	нет	да	да	да
Отзыв заявки	да	да	да	да
Восстановление отозванной заявки	да, если отозвана им же	да, если отозвана им же	да	да
Редактирование темы заявки	да	да	да	да
Редактирование своих контактных данных	да	да	да	да
Обработка запросов на изменение квоты	нет	нет	да	да
Экспорт реестра	нет	нет	да	да
Генерация проекта приказа	нет	нет	да	да
Управление пользователями	нет	нет	нет	да
Управление справочниками	нет	нет	нет	да

Таблица Б.3 – Сценарий использования UC-01 «Подача заявки студентом»

Поле	Значение
ID	UC-01
Название	Подача заявки на руководство ВКР
Актор	Студент
Предусловие	Студент авторизован в системе; руководитель имеет вакантные места на руководство ВКР
Постусловие	Создана новая заявка со статусом «Ожидает»; журнал истории заявки пополнен записью о создании заявки студентом.
Основной поток	1. Студент кликает на кнопку «Подать заявку». 2. Система отображает список руководителей с указанием количества вакантных мест на руководство ВКР. 3. Студент выбирает руководителя. 4. Система проверяет, есть ли у выбранного руководителя вакантные места на руководство ВКР. 5. Система проверяет, не подавал ли данный студент заявку этому руководителю ранее. 6. Система открывает форму подачи заявки. 7. Студент вводит тему и описание ВКР, используемые технологии и подтверждает создание заявки. 8. Система создаёт заявку со статусом «Ожидает». 9. Система добавляет запись в историю заявки с указанием времени действия, пользователя (студента) и действия. 10. Система отображает сообщение об успешном создании заявки.
Альтернативный поток A1 (превышение нагрузки)	На шаге 4 система определяет, что у выбранного руководителя нет вакантных мест на руководство ВКР. Студенту выводится сообщение о невозможности подачи заявки данному руководителю. Сценарий завершается.
Альтернативный поток A2 (дублирующая заявка)	На шаге 5 система обнаруживает, что у данного студента уже существует заявка этому руководителю. Студенту выводится сообщение о невозможности подачи заявки данному руководителю. Сценарий завершается.

Таблица Б.4 – Сценарий использования УС-02 «Одобрение заявки руководителем»

Поле	Значение
ID	УС-02
Название	Одобрение заявки руководителем
Актор	Руководитель
Предусловие	Руководитель авторизован в системе; существует хотя бы одна заявка со статусом «Ожидает», адресованная данному руководителю
Постусловие	Статус заявки изменён на «Одобрена»; количество закреплённых за руководителем студентов увеличено на единицу; студент видит обновлённый статус; в журнал истории заявки добавлена запись об одобрении заявки руководителем
Основной поток	1. Система отображает список всех заявок, адресованных руководителю, со статусом «Ожидает». 2. Руководитель выбирает заявку и нажимает кнопку «Одобрить». 3. Система изменяет статус заявки на «Одобрена».
Основной поток	4. Система уменьшает количество вакантных мест руководителя на руководство ВКР на единицу. 5. Система добавляет запись в историю заявки с указанием времени действия, пользователя (руководителя) и действия. 6. Система отображает сообщение об успешном одобрении заявки.

Таблица Б.5 – Сценарий использования UC-03 «Формирование проекта приказа заведующим кафедрой»

Поле	Значение
ID	UC-03
Название	Формирование проекта приказа заведующим кафедрой
Актор	Заведующий кафедрой
Предусловие	Заведующий кафедрой авторизован в системе; существует хотя бы одна заявка со статусом «Одобрена»
Постусловие	Сгенерирован файл документа в формате Word, оформленный по шаблону университета
Основной поток	1. Заведующий кафедрой переходит в приложение «Заявки». 2. Заведующий кафедрой нажимает кнопку «Сформировать приказ». 3. Заведующий кафедрой выбирает учебную группу из выпадающего списка и нажимает кнопку «Сформировать». 4. Система выполняет запрос к базе данных для получения всех заявок выбранной группы со статусом «Одобрено». 5. Система загружает шаблон приказа в формате Word. 6. Система подставляет данные из каждой заявки (ФИО студента, тема ВКР, ФИО и данные руководителя) в соответствующие поля шаблона. 7. Система формирует файл и передаёт его для скачивания. 8. Заведующий кафедрой сохраняет файл на своё устройство.

Таблица Б.6 – Сценарии тестирования критических путей пользователей

ID	Название	Роль	Предусловие	Шаги	Ожидаемый результат	Результат
1	2	3	4	5	6	7
TK-01	Подача заявки руководителю, имеющему вакантные места	Студент	Студент авторизован; у руководителя есть вакантные места	1. Перейти к списку руководителей. 2. Выбрать руководителя. 3. Ввести тему, описание и технологии. 4. Нажать «Подать заявку».	Заявка создана; статус «Ожидает»; руководитель видит заявку в своём списке	Выполнено успешно
TK-02	Подача заявки руководителю, не имеющему вакантных мест	Студент	Студент авторизован; у руководителя нет вакантных мест	1. Перейти к списку руководителей. 2. Выбрать руководителя. 3. Попытаться подать заявку.	Заявка не создана; выведено сообщение: «Выбранный руководитель не имеет вакантных мест на руководство ВКР»	Выполнено успешно
TK-03	Защита от дублирования заявок	Студент	Студент авторизован; уже существует заявка этому руководителю	Попытаться подать вторую заявку тому же руководителю	Система блокирует создание; выведено сообщение о существующей заявке.	Выполнено успешно
TK-04	Одобрение заявки руководителем	Руководитель	Руководитель авторизован; есть входящая заявка со статусом «Ожидает»	1. Перейти на страницу входящих заявок. 2. Кликнуть на «Одобрить».	Статус изменён на «Одобрена»; количество закреплённых за руководителем студентов увеличено на единицу; студент видит изменение статуса	Выполнено успешно
TK-05	Отклонение заявки руководителем	Руководитель	Руководитель авторизован; есть входящая заявка со статусом «Ожидает»	1. Перейти на страницу входящих заявок. 2. Кликнуть на «Отклонить».	Статус изменён на «Отклонена»; количество закреплённых за руководителем студентов не изменилось; студент видит изменение статуса	Выполнено успешно

Продолжение таблицы Б.6

1	2	3	4	5	6	7
ТК-06	Отзыв заявки студентом	Студент	Студент авторизован; есть заявка в статусе «Ожидает».	1. Перейти на страницу «Мои заявки». 2. Кликнуть на «Отозвать».	Статус изменён на «Отозвана»; руководитель видит изменение статуса	Выполнено успешно
ТК-07	Отзыв одобренной заявки руководителем	Руководитель	Руководитель авторизован; есть одобренная заявка	1. Перейти на страницу входящих заявок. 2. Кликнуть на «Отозвать».	Статус изменён на «Отозвана»; количество закреплённых за руководителем студентов уменьшено на 1; студент видит изменение статуса	Выполнено успешно
ТК-08	Восстановление отозванной заявки	Студент	Студент авторизован; заявка была отозвана студентом и имела статус «Одобрена».	1. Перейти на страницу «Мои заявки». 2. Кликнуть на «Восстановить».	Статус восстановлен на «Одобрена»; количество закреплённых за руководителем студентов увеличено на 1; руководитель видит изменение статуса	Выполнено успешно
ТК-09	Запрос изменения квоты	Руководитель	Руководитель авторизован	1. Нажать «Запросить изменение квоты». 2. Ввести желаемое значение и обоснование. 3. Отправить запрос	Запрос создан со статусом «Ожидает рассмотрения» и отображается у заведующего кафедрой	Выполнено успешно
ТК-10	Одобрение запроса на изменение квоты	Заведующий кафедрой	Заведующий кафедрой авторизован; есть запрос от руководителя	1. Перейти в приложение «Профили», раздел запросов. 2. Выбрать запрос и кликнуть на «Одобрить».	Статус запроса изменён на «Одобрена»; квота руководителя обновлена	Выполнено успешно

Продолжение таблицы Б.6

1	2	3	4	5	6	7
TK-11	Формирование проекта приказа	Заведующий кафедрой	Заведующий кафедрой авторизован; есть одобренные заявки.	1. Перейти в список заявок. 2. Кликнуть на «Сформировать приказ». 3. Выбрать учебную группу из выпадающего списка. 4. Кликнуть «Сформировать».	Сгенерирован файл в формате Word; форматирование соответствует шаблону университета.	Выполнено успешно
TK-12	Экспорт реестра в Excel	Заведующий кафедрой	Заведующий кафедрой авторизован; есть заявки в системе.	1. Перейти в приложение «Заявки». 2. Кликнуть на «Экспорт в Excel».	Сгенерирован файл .xlsx; данные соответствуют текущим фильтрам.	Выполнено успешно
TK-13	Ведение журнала изменений	Система	Заявка создана и изменяла статус.	Проверить поле history в записи заявки.	В журнале зафиксированы все действия с указанием времени, пользователя и типа действия.	Выполнено успешно

ПРИЛОЖЕНИЕ В

Листинги

Листинг В.1 – Модель заявки ThesisApplication

```
1 ...
2 class ThesisApplication(models.Model):
3     STATUS_CHOICES = (
4         ('sent', 'Ожидает'),
5         ('approved', 'Одобрена'),
6         ('rejected', 'Отклонена'),
7         ('withdrawn', 'Отозвана'),
8     )
9
10    student = models.ForeignKey(Student, on_delete=...
11    professor = models.ForeignKey(Professor, on_delete=...
12    topic = models.TextField(verbose_name='Тема работы')
13    description = models.TextField(blank=True, null=True)
14    technologies = models.ManyToManyField('technologies...
15    technologies_other = models.CharField(max_length=255...
16    status = models.CharField(max_length=20, choices=...
17    previous_status = models.CharField(max_length=20, ...
18    withdrawn_by = models.CharField(max_length=20, ...
19    history = models.JSONField(default=list, blank=True)
20    created_at = models.DateTimeField(auto_now_add=True)
21    updated_at = models.DateTimeField(auto_now=True)
22
23    class Meta:
24        unique_together = ['student', 'professor']
25        verbose_name = 'Заявка'
26        verbose_name_plural = 'Заявки'
27 ...
```

ПРИЛОЖЕНИЕ Г

Первичный запрос к нейросетевой модели

Сгенерируй исполняемый программный код на Python для наполнения базы данных тестовыми данными. Код должен создавать записи для трёх кафедр («Инфокогнитивные технологии», «Прикладная информатика», «Информационная безопасность») Факультета информационных технологий: 180 студентов (20 в каждой из 9 групп), 24 преподавателя (по 8 на кафедре). У каждого студента должно быть 1 или 2 заявки со статусами (60% одобренных, 20% в ожидании, 10% отклонённых, 10% отозванных). У всех уникальные ФИО. У преподавателей квота от 10 до 12. Первый преподаватель на кафедре это заведующий. Логины формируются как транслитерация фамилии + инициалы, пароль для всех пользователей это «test123». Технологии: «Веб-фреймворки и анализ данных», «Корпоративная разработка», «Сетевые технологии и безопасность».

ПРИЛОЖЕНИЕ Д

Техническое задание

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
**«МОСКОВСКИЙ ПОЛИТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»
(МОСКОВСКИЙ ПОЛИТЕХ)**

УТВЕРЖДАЮ

Руководитель образовательной программы

_____/ М.В. Даньшина
Дата: «____» _____ 20_г.

УТВЕРЖДАЮ

Заведующий кафедрой

_____/ Е.А. Пухова
Дата: «____» _____ 20_г.

Автоматизированная информационная система

по теме:

ВЕБ-ПРИЛОЖЕНИЕ ДЛЯ АВТОМАТИЗАЦИИ УПРАВЛЕНИЯ ВЫПУСКНЫМИ КВАЛИФИКАЦИОННЫМИ РАБОТАМИ

ТЕХНИЧЕСКОЕ ЗАДАНИЕ

На 4 листах

Действует с «____» _____ 20

СОГЛАСОВАНО

Руководитель ВКР

_____/ В. В. Осьмин
Дата: «____» _____ 20_г

Москва 2026

1. ОБЩИЕ СВЕДЕНИЯ

1.1. Цели создания АС

Целью разработки является автоматизация процесса управления выпускными квалификационными работами: от подачи заявки студентом до формирования проекта приказа заведующим кафедрой, с обеспечением учёта вакантных мест у руководителей и информированности всех участников о ходе распределения студентов.

1.2 Перечень документов, на основании которых создаётся АС

Приказ от 05.05.2026 №2491 – УД «О назначении руководителей и утверждении тем выпускных квалификационных работ».

1.3 Плановые сроки начала и окончания работы

Плановый срок начала работы по созданию веб-приложения – 2 февраля 2026 г. Плановый срок окончания работы по созданию веб-приложения – 2 июня 2026 г.

2. ЦЕЛИ И НАЗНАЧЕНИЕ СОЗДАНИЯ АВТОМАТИЗИРОВАННОЙ СИСТЕМЫ

2.1. Цели создания АС

Разработка веб-приложения, позволяющего: студентам выбирать научного руководителя на основе актуальной информации о вакантных местах и подавать заявки; руководителям управлять входящими заявками и запрашивать изменение квоты; заведующим кафедрами контролировать процесс, обрабатывать запросы, формировать реестры и проекты приказов; администраторам вести справочники системы.

2.2. Назначение АС

Система предназначена для автоматизации деятельности кафедры университета в части распределения студентов по научным руководителям и утверждения тем ВКР. Система обеспечивает: единое информационное пространство для всех участников процесса; автоматический учёт вакантных мест у руководителей; двусторонний отзыв и восстановление заявок; генерацию выходных документов (реестр Excel, проект приказа Word по шаблону университета).

3. ОСНОВАНИЯ ДЛЯ РАЗРАБОТКИ

Задание на выпускную квалификационную работу по направлению 09.03.03 «Прикладная информатика», образовательная программа «Информационные технологии управления бизнесом», утверждённое заведующим кафедрой «Инфокогнитивные технологии».

4. ТРЕБОВАНИЯ К ВЕБ-ПРИЛОЖЕНИЮ

Объектом автоматизации является процесс распределения студентов выпускных курсов по научным руководителям и утверждения тем выпускных квалификационных работ на кафедре «Инфокогнитивные технологии» Московского политехнического университета. В рамках данного процесса участвуют студенты, научные руководители, заведующий кафедрой и администратор. В настоящее время процесс ведётся с использованием электронных таблиц и средств электронной почты, что приводит к высокой трудоёмкости, ошибкам ручного ввода и недостаточной информированности участников. Условия эксплуатации создаваемой автоматизированной системы – штатное функционирование в сети кафедры или факультета с доступом пользователей через веб-браузер.

5. ТРЕБОВАНИЯ К ВЕБ-ПРИЛОЖЕНИЮ

5.1. Требования к структуре в целом

Система состоит из клиентской части (веб-интерфейс на HTML, CSS, JavaScript с использованием библиотеки стилей Bootstrap), серверной части на базе программной платформы Django и системы управления базами данных SQLite с возможностью миграции на промышленную СУБД. Взаимодействие между компонентами осуществляется по протоколу HTTP/HTTPS.

5.2. Требования к функциям

Система должна обеспечивать авторизацию пользователей с разграничением четырёх ролей: студент, руководитель, заведующий кафедрой, администратор. Студент должен иметь возможность просматривать список руководителей своей кафедры с отображением количества вакантных мест на руководство ВКР и области научных интересов, подавать заявку выбранному руководителю с указанием темы, описания и используемых технологий, а также отзывать и восстанавливать свою заявку. Руководитель должен иметь возможность просматривать входящие заявки, одобрять или отклонять их, запрашивать изменение своей квоты на руководство ВКР с обоснованием, а также отзывать и восстанавливать заявки, которые он отклонил или отозвал. Заведующий кафедрой должен иметь возможность просматривать все данные по своей кафедре, обрабатывать запросы руководителей на изменение квоты, экспортировать реестр заявок в формате Excel и генерировать проект приказа об утверждении тем и назначении руководителей в формате Word по шаблону университета с группировкой по учебным группам. Администратор должен иметь возможность управлять справочниками системы.

5.3. Требования к информационному обеспечению

Система должна хранить данные о пользователях, профилях студентов и руководителей, заявках на руководство ВКР (с указанием темы, описания, технологий, статуса, журнала изменений и временных меток), запросах на изменение квоты, а также справочные данные о факультетах, кафедрах, профилях образования, должностях, учёных степенях, учёных званиях, технологиях и категориях технологий. Взаимодействие с базой данных осуществляется через средства объектно-реляционного отображения Django.

6. СОСТАВ И СОДЕРЖАНИЕ РАБОТ ПО СОЗДАНИЮ АВТОМАТИЗИРОВАННОЙ СИСТЕМЫ

Разработка системы включает следующие этапы: анализ предметной области и существующего процесса управления ВКР; обзор и сравнительный анализ существующих программных решений; формулирование требований к системе; проектирование бизнес-процессов в нотациях BPMN, IDEF0 и EPC; проектирование архитектуры и выбор технологий; проектирование структуры базы данных; реализация серверной части на программной платформе Django; реализация клиентской части с использованием библиотеки стилей Bootstrap; проведение функционального тестирования; оценка эффективности разработанной системы; оформление пояснительной записки.

7. ПОРЯДОК РАЗРАБОТКИ АС

Разработка ведётся в соответствии с календарным планом, приведённым в задании на выпускную квалификационную работу, и включает последовательное выполнение этапов, указанных в разделе 5 настоящего технического задания.

8. ПОРЯДОК КОНТРОЛЯ И ПРИЕМКИ АС

Приёмка системы осуществляется на основе результатов функционального тестирования. Перечень тест-кейсов приведён в таблице Б.6 пояснительной записки. Все тест-кейсы должны быть выполнены успешно. Результаты тестирования фиксируются в той же таблице. Система считается принятой, если она обеспечивает выполнение всех функциональных требований, указанных в разделе 5.2.

9. ТРЕБОВАНИЯ К ДОКУМЕНТИРОВАНИЮ

В состав документации входят пояснительная записка к выпускной квалификационной работе и техническое задание.

10. ПОДГОТОВКА ОБЪЕКТА АВТОМАТИЗАЦИИ К ВВОДУ В ДЕЙСТВИЕ

Для ввода системы в эксплуатацию необходимо развернуть веб-приложение на сервере кафедры, установить зависимости, выполнить миграции базы данных, загрузить начальные справочные данные через административную панель и провести инструктаж сотрудников кафедры по работе с системой.

11. ИСТОЧНИКИ РАЗРАБОТКИ

ГОСТ 34.602-2020 «Информационные технологии. Комплекс стандартов на автоматизированные системы. Техническое задание на создание автоматизированной системы». Материалы выпускной квалификационной работы.