

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное бюджетное образовательное учреждение высшего образования

«ПЕНЗЕНСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»

Факультет

вычислительной техники

Кафедра

«Вычислительная техника»

Специальность: 09.05.01 «Применение и эксплуатация
автоматизированных систем специального
назначения»
Специализация: «Эксплуатация вычислительных машин,
комплексов, систем и сетей специального
назначения»

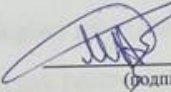
ДИПЛОМНЫЙ ПРОЕКТ

на тему

МОДУЛЬ ПРОВЕДЕНИЯ МНОГОЭТАПНОГО ОТБОРА
АВТОМАТИЗИРОВАННОЙ СИСТЕМЫ УПРАВЛЕНИЯ

ПЕРСОНАЛОМ

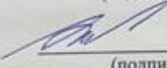
Студент

 11.06.25 Манжурин Андрей Владимирович
(подпись, дата) (ФИО полностью)

Руководитель

 11.06.25 Бикташев Р.А.
(подпись, дата) (фамилия, инициалы)

Нормоконтролёр

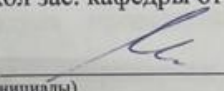
 12.06.25 Бычков А.С.
(подпись, дата) (фамилия, инициалы)

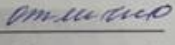
Рецензент

Исх. Управление 17  13.06.25 Себастьян И.К.
(должность, место работы) (подпись, дата) (фамилия, инициалы)

Работа допущена к защите (протокол зас. кафедры от 28.05.2025 г. № 16)

Заведующий кафедрой

 Митрохин М.А.
(подпись) (фамилия, инициалы)

Работа защищена с отметкой 

(протокол заседания ГЭК от 19.06.2025 г. № 7)

Секретарь ГЭК

 (подпись) (фамилия, инициалы)

Пенза 2025

Федеральное государственное бюджетное образовательное учреждение
высшего образования
ПЕНЗЕНСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
ПОЛИТЕХНИЧЕСКИЙ ИНСТИТУТ

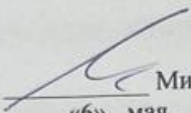
Факультет

Вычислительной техники

Кафедра

Вычислительная техника

«Утверждаю»
Заведующий кафедрой

 Митрохин М.А.
«6» мая 2025 г.

ЗАДАНИЕ НА ДИПЛОМНЫЙ ПРОЕКТ

1. Студент Манжури́н Андрей Владимирович гр. 20ВОЭ1 факультета вычислительной техники.

2. Тема работы: «Модуль проведения многоэтапного отбора автоматизированной системы управления персоналом».

Тема утверждена приказом ПГУ № 601/о от 06.05.2025 г.

3. Руководитель работы к.т.н., профессор каф. ВТ «ПГУ» Бикташев Р.А.

4. Задание на работу (назначение разработки, исходные данные и т.п.).

Разработать модуль проведения многоэтапного отбора автоматизированной системы управления персоналом. Автоматизированная система управления персоналом должна выполнять следующие функции: подача заявок, формирование команд, заполнение чек-листов, управление этапами заявки, учет данных о заявках. Модуль проведения многоэтапного отбора автоматизированной системы управления персоналом должна разрабатываться с помощью ПО Websoft HCM. База данных должна разрабатываться для СУБД «MS SQL».

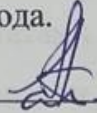
5. Перечень подлежащих разработке вопросов.

Требуется произвести анализ предметной области, определить функциональный состав программы, выбрать стек технологий для реализации программы, разработать алгоритм работы программы, разработать диаграммы: прецедентов, активностей, последовательностей, объектов, классов, с последующим написанием программы.

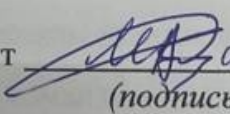
6. Календарный график выполнения работы

№ п/п	Наименование этапов работы	Объем работы	Срок выполнения	Подпись руководителя
1.	Анализ предметной области	10	06.05.2025 - 09.05.2025	
2.	Проектирование программного приложения	10	10.05.2025 - 13.05.2025	
3.	Разработка базы данных	20	14.05.2025 - 18.05.2025	
4.	Отладка и тестирование	10	19.05.2025 - 22.05.2025	
5.	Разработка экономической и бжд части	10	23.05.2025 - 27.05.2025	
6.	Оформление графической части	10	28.05.2025	
7.	Оформление выпускной работы	30	29.05.2025 - 10.06.2025	

Дата выдачи задания « 6 » мая 2025 года.

Руководитель дипломного проекта  06.05.25 Бикташев Р.А.

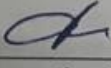
(подпись, дата)

Задание к исполнению принял студент  06.05.25 Манжурин А.В.

(подпись, дата)

Работу к защите допустить

Декан факультета

 18.06.2025 Фионова Л.Р.

(подпись, дата)

Реферат

Пояснительная записка содержит 146 листов, 19 рисунков, 15 таблиц, 10 источников и 4 приложения.

МОДУЛЬ МНОГОЭТАПНОЙ ОТБОРА ПЕРСОНАЛА, WEBSOFT HCM, JAVASCRIPT, REACT, MS SQL SERVER, КЛИЕНТ-СЕРВЕРНАЯ АРХИТЕКТУРА, БАЗА ДАННЫХ

Цель работы – разработать модуль для автоматизации многоэтапной оценки компетенций сотрудников в системе управления персоналом Websoft HCM.

Объекты разработки:

- серверное приложение на платформе Websoft, реализующее бизнес-логику модуля: управление заявками, командами, оценками, интеграцию с базой данных и документооборотом;

- клиентское приложение на React.js с интерфейсом для взаимодействия пользователей, включая подачу заявок, формирование команд, заполнение чек-листов и отслеживание этапов оценки.

В результате проделанной работы разработан программный модуль, полностью соответствующий поставленной задаче. Разработана серверная часть на платформе Websoft HCM, реализующая бизнес-логику по управлению заявками, командами и результатами оценки. Реализовано API для взаимодействия между клиентом и сервером на основе RPC-паттерна. Разработана клиентская часть на React.js с интуитивно понятным графическим интерфейсом, обеспечивающим взаимодействие пользователей с системой: подача заявок, управление командами, заполнение чек-листов и отслеживание этапов многоэтапной оценки. Все данные сохраняются в реляционную базу данных MS SQL Server с обеспечением целостности и безопасности.

					ПГУ1.090501.12.001 ПЗ				
Изм.	Лист	№ докум.	Подпись	Дата	Модуль проведения многоэтапного отбора автоматизированной системы управления персоналом Пояснительная записка	Лит.	Лист	Листов	
Разраб.		Манжури А.В.		11.06.25					
Провер.		Бикташев Р.А.		11.06.25			2	146	
Н. Контр.		Бычков А.С.		12.06.25					
Утверд.		Митрохин М.А.		11.06.25					
						ФВТ, каф. ВТ гр.20ВОЭ1			

Содержание

Перечень сокращений и обозначений.....	5
Введение.....	6
1 Модуль проведения многоэтапного отбора автоматизированной системы управления персоналом	8
1.1 Анализ предметной области	8
1.1.1 Актуальность разработки.....	8
1.1.2 Анализ существующих аналогов	9
1.1.3 Обоснование актуальности разработки программы.....	11
1.2 Выбор технологий и средств разработки приложения	15
1.2.1 Платформа Websoft HCM.....	15
1.2.2 Выбор языка программирования.....	16
1.2.3 Интерфейс и визуализация.....	17
1.2.4 База данных и работа с данными.....	18
1.2.5 Среда разработки	19
1.2.6 Обмен данными JSON	19
1.3 Проектирование информационной системы	20
1.3.1 Паттерн RPC	20
1.3.2 Архитектура модуля	21
1.3.3 Проектирование базы данных	24
1.4 Разработка приложения	29
1.4.1 Хранение данных	29
1.4.2 Описание программы	30
1.4.3 Руководство пользователя	37
1.4.4 Методика испытаний программы	44
2 Техничко-экономическое обоснование разработки	48
2.1 Введение.....	48
2.2 Расчет затрат на разработку программы.....	49
2.3 Расчет цены разработанной программы	50
2.4 Расчет капитальных вложений	50

2.5 Расчет эксплуатационных расходов.....	51
2.6 Расчет денежного годового экономического эффекта.....	53
2.7 Итоговые расходы.....	54
3 Охрана труда.....	56
Заключение	60
Список использованных источников	61
Приложение А – UML-диаграммы	62
Приложение Б – Листинг программы	66
Приложение В – Результаты испытаний	113
Приложение Г – Презентация	140

Перечень сокращений и обозначений

АС – автоматизированная система

БД – база данных

СУБД – система управления базами данных

API – Application Programming Interface (интерфейс программного взаимодействия)

FK – Foreign Key (внешний ключ)

GUI – Graphical User Interface (графический интерфейс пользователя)

HCM – Human Capital Management (управление человеческим капиталом)

IDE – Integrated Development Environment (интегрированная среда разработки)

JSON – JavaScript Object Notation (формат обмена данными)

MS SQL Server – Microsoft Structured Query Language Server

PK – Primary Key (первичный ключ)

RPC – Remote Procedure Call (удалённый вызов процедур)

SaaS – Software as a Service (программное обеспечение как услуга)

SQL – Structured Query Language (язык структурированных запросов)

UI – User Interface (пользовательский интерфейс)

UML – Unified Modeling Language (унифицированный язык моделирования)

VS Code – Visual Studio Code

					ПГУ 1.090501.12.001 ПЗ	Лист
						5
Изм.	Лист	№ докум.	Подпись	Дата		

Введение

В условиях стремительного развития цифровых технологий и постоянного роста конкуренции между компаниями, особое значение приобретает эффективное управление человеческими ресурсами. Персонал становится не просто исполнителем задач, а стратегическим ресурсом, определяющим динамику роста и инновационную способность организации. Компании все чаще обращаются к современным информационным технологиям для автоматизации рутинных HR-процессов и обеспечения прозрачности при принятии управленческих решений. Одним из ключевых направлений автоматизации является процесс подбора персонала, который, несмотря на его важность, во многих организациях до сих пор осуществляется вручную или с использованием разрозненных программных решений.

Современные методы подбора требуют не только оценки соответствия кандидата квалификационным требованиям, но и всестороннего анализа его личностных качеств, коммуникативных навыков, способности работать в команде, потенциала к развитию. Для этого применяются многоэтапные схемы отбора, включающие онлайн-тестирование, интервью, кейсовые задания, оценки по компетенциям и другие методы. Однако в случае отсутствия централизованной автоматизированной системы такие процессы становятся трудоёмкими и подверженными ошибкам, что снижает эффективность работы отдела персонала и увеличивает срок закрытия вакансий.

Одним из эффективных решений является создание модуля в составе корпоративной HRM-системы, обеспечивающего автоматизированное проведение всех этапов подбора и предоставляющего инструмент для объективной и комплексной оценки кандидатов. Подобный подход позволяет сократить время отклика на заявку, стандартизировать процесс оценки, снизить субъективность принимаемых решений, а также упростить взаимодействие между HR-отделом и руководителями подразделений.

В данной выпускной квалификационной работе рассматривается

					ПГУ 1.090501.12.001 ПЗ	Лист
						6
Изм.	Лист	№ докум.	Подпись	Дата		

разработка модуля многоэтапного отбора в составе системы управления персоналом Websoft HCM. Разрабатываемая система ориентирована на поддержку ключевых этапов рекрутинга: от подачи заявки кандидатом до вынесения решения о найме. Особенностью модуля является использование оценки по компетенциям и возможность гибкой настройки этапов в соответствии с корпоративными требованиями. Также предусмотрено взаимодействие между различными ролями пользователей – кандидатами, HR-специалистами, руководителями подразделений – с сохранением прозрачности и логики в бизнес-процессах.

Актуальность работы обусловлена растущей потребностью организаций в цифровизации HR-процессов, необходимостью повышения точности и скорости принятия кадровых решений, а также потребностью в индивидуальной адаптации систем под структуру и задачи конкретного предприятия. Наиболее остро эта проблема стоит в средних и крупных компаниях, в которых численность персонала и объем рекрутинговых процессов требуют внедрения гибких и масштабируемых решений.

Целью данной работы является создание программного модуля, интегрируемого в существующую HCM-систему, для автоматизации процессов многоэтапного отбора. В работе проведён анализ существующих решений, выявлены ключевые проблемы их применения, сформулированы функциональные и технические требования к модулю, реализована архитектура системы, выполнено тестирование и предложены пути дальнейшего развития.

					ПГУ 1.090501.12.001 ПЗ	Лист
						7
Изм.	Лист	№ докум.	Подпись	Дата		

1 Модуль проведения многоэтапного отбора автоматизированной системы управления персоналом

1.1 Анализ предметной области

1.1.1 Актуальность разработки

Эффективное управление человеческими ресурсами является важнейшей составляющей устойчивого развития любой организации. Особую значимость приобретает качественный подбор персонала, особенно в крупных компаниях и структурах с высоким уровнем ответственности – государственных учреждениях, силовых ведомствах, оборонных предприятиях, а также коммерческих организациях, работающих в условиях высокой конкуренции. Успешность кадровой политики напрямую зависит от способности предприятия проводить всестороннюю, объективную и своевременную оценку кандидатов при найме на работу.

На практике процесс отбора персонала представляет собой многоэтапную процедуру, включающую тестирование, интервьюирование, анкетирование, оценку компетенций, проверку биографических данных и другие этапы. При ручной организации подобных процессов возникают значительные затраты времени и ресурсов, возрастает риск человеческой ошибки, снижается прозрачность и объективность оценки кандидатов. Кроме того, невозможность оперативной обработки большого объема данных делает невозможным масштабирование процедуры отбора на широкую аудиторию претендентов.

В современных условиях эти проблемы решаются за счёт внедрения автоматизированных систем управления персоналом, одной из ключевых задач которых становится цифровизация процедуры отбора. Разработка и внедрение специализированного модуля, обеспечивающего автоматическое проведение многоэтапного отбора кандидатов, позволяет не только ускорить процесс найма, но и повысить его точность, обеспечить единые стандарты оценки, снизить нагрузку на кадровые службы, а также минимизировать

					ПГУ 1.090501.12.001 ПЗ	Лист
						8
Изм.	Лист	№ докум.	Подпись	Дата		

субъективность при принятии решений.

Предлагаемая в данной выпускной квалификационной работе система направлена на решение актуальной задачи – автоматизацию многоэтапного процесса подбора кадров в рамках комплексной системы управления личным составом. Разработка обеспечит возможность гибкой настройки этапов отбора, интеграции с внешними источниками данных, хранения и анализа результатов тестов и анкет, формирования отчетов, а также взаимодействия с соискателями через цифровой интерфейс. Таким образом, проект имеет высокую практическую значимость и отвечает актуальным потребностям в области автоматизации HR-процессов, особенно в организациях с высоким кадровым оборотом и строгими требованиями к качеству подбора персонала.

1.1.2 Анализ существующих аналогов

В рамках разработки автоматизированной системы управления личным составом особое внимание уделяется модулю проведения многоэтапного отбора, включающему оценку компетенций сотрудников и команд. Такой подход позволяет не только отбирать кандидатов по формальным критериям, но и учитывать их соответствие корпоративной культуре, профилю должности, а также способности к эффективной командной работе.

Для обоснования актуальности разработки и выбора архитектурных решений был проведён анализ существующих программных продуктов, решающих схожие задачи.

Websoft HCM.

Выступает в качестве базовой платформы для создаваемого решения. Обладает развитой модульной структурой и возможностью гибкой настройки бизнес-процессов. Однако в типовой поставке отсутствует специализированный инструмент, реализующий именно многоэтапный отбор с оценкой командных и личностных компетенций, а также гибкую настройку этапов с возможностью интеграции кастомных страниц. Следовательно, создание такого модуля представляет собой логичное и востребованное направление развития платформы.

					ПГУ 1.090501.12.001 ПЗ	Лист
						9
Изм.	Лист	№ докум.	Подпись	Дата		

SAP SuccessFactors.

Один из лидеров на международном рынке HR-автоматизации. Включает мощный набор инструментов оценки компетенций, построенных на базе машинного обучения, с возможностью индивидуальной настройки этапов найма и автоматической генерации рекомендаций. Однако данная система в значительной степени ориентирована на западные модели компетенций, что может ограничивать её применимость в российских реалиях. Кроме того, высокая стоимость внедрения и локализации делает её недоступной для ряда организаций.

Talantix.

Российское облачное решение, ориентированное на автоматизацию массового подбора. Содержит базовые механизмы подбора, такие как этапы найма, воронка откликов, интеграция с job-сайтами. Однако отсутствует оценка командной совместимости, глубокая кастомизация процессов и внедрение пользовательских страниц. Система подходит для стандартных сценариев, но не гибка для адаптации под сложные модели оценки.

1С:Зарплата и управление персоналом

Распространённое решение в российских организациях, однако основной функционал сосредоточен на кадровом и расчётном учёте. Механизмы подбора и оценки персонала представлены слабо, отсутствует полноценная поддержка многоэтапного отбора, оценки компетенций и настройка пользовательских элементов интерфейса.

Harver.

Международное SaaS-решение, предназначенное для массового подбора. Использует онлайн-оценки soft skills, поведенческих особенностей и когнитивных способностей. Однако не предоставляет возможностей оценки командного взаимодействия, ограничен в адаптации под внутренние процессы, не поддерживает расширенную настройку страниц или этапов под конкретного заказчика.

Сравнительный анализ представленных решений обобщён в таблице

					ПГУ 1.090501.12.001 ПЗ	Лист
						10
Изм.	Лист	№ докум.	Подпись	Дата		

1.1.2, где приведены ключевые характеристики систем с точки зрения их применимости для реализации многоэтапного отбора с оценкой компетенций.

Таблица 1.1.2 – Сравнительный анализ аналогов

Система	Оценка компетенций	Гибкость настройки	Кастомные страницы	Стоимость внедрения
Websoft HCM	Да	Высокая	Да	Средняя
SAP SuccessFactors	Да	Высокая	Да	Высокая
Talantix	Частично	Средняя	Нет	Низкая
1С:ЗУП	Нет	Низкая	Нет	Средняя
Harver	Да	Низкая	Нет	Высокая

1.1.3 Обоснование актуальности разработки программы

Разработка модуля для интеграции в систему Websoft HCM обусловлена уникальными потребностями компаний в автоматизации комплексной оценки персонала. Использование инфраструктуры Websoft HCM обеспечивает конкурентное ценовое преимущество, исключая затраты на создание автономных решений и сокращая расходы на внедрение и сопровождение. Интеграция позволяет задействовать существующие HR-процессы и расширяет функциональность системы без дублирования компонентов.

Многоэтапная оценка компетенций – ключевая инновация модуля, решающая проблему поверхностного анализа навыков. Она обеспечивает:

- глубину анализа за счёт поэтапной проверки (чек-листы) с накоплением результатов на каждом этапе;
- объективность через средневзвешенный расчёт баллов, минимизирующий субъективность;
- повышение эффективности персонала путём выявления слабых зон для индивидуальных программ развития, что напрямую влияет на будущую работоспособность сотрудников.

Для бизнеса это трансформируется в конкретные преимущества:

- стандартизацию оценки для тысяч сотрудников;
- рост вовлечённости через геймификацию;

– снижение операционных рисков благодаря автоматизации рутинных процессов.

Модуль трансформирует ресурсоёмкую ручную оценку в масштабируемый технологичный процесс, напрямую влияющий на качество персонала и бизнес-результаты через точные метрики, снижение затрат и синергию с корпоративной HR-экосистемой.

1.1.4 Постановка задачи

Требуется реализовать модуль, который будет соответствовать следующим функциональным требованиям:

– возможность управления заявками на оценку (подача индивидуальных и групповых заявок с выбором номинации, автоматическое распределение заявок по этапам оценки на основе заданных правил, организация документооборота заявок с поддержкой последовательных этапов: проверка директором подразделения, оценка территориальным управляющим, финальное судейство, трекинг и отображение актуального статуса заявки на каждом этапе);

– возможность управления командами участников (создание, редактирование и расформирование команд, обеспечение ограничений по количеству участников в команде - 4);

– возможность проведения оценки компетенций (предоставление автоматизированных чек-листов для оценщиков, автоматическое сохранение промежуточных и финальных результатов оценки, предоставление инструментов анализа данных);

– возможность управления доступом (настройка ролевого доступа к функциям системы в зависимости от должности и подразделения пользователя, ограничение видимости данных и доступных действий на основе роли пользователя);

– возможность работы с системой коммуникаций и уведомлений (отправка персонализированных уведомлений участникам оценки, руководителям и судьям, автоматические напоминания о сроках выполнения действий и этапов, оповещения об изменениях статусов заявок и доступности

					ПГУ 1.090501.12.001 ПЗ	Лист
						12
Изм.	Лист	№ докум.	Подпись	Дата		

результатов оценки);

- возможность интеграции и обеспечения безопасности данных (интеграция с порталом Websoft HCM для реализации единой авторизации, привязка всех объектов системы (заявки, команды, оценки, результаты) к конкретному году проведения оценки, ограничение доступа к данным на основе временных интервалов (доступ только к данным текущего/активного года, архивация данных прошлых лет).

Диаграмма вариантов использования модуля на этапе регистрации представлен на рисунке 1.

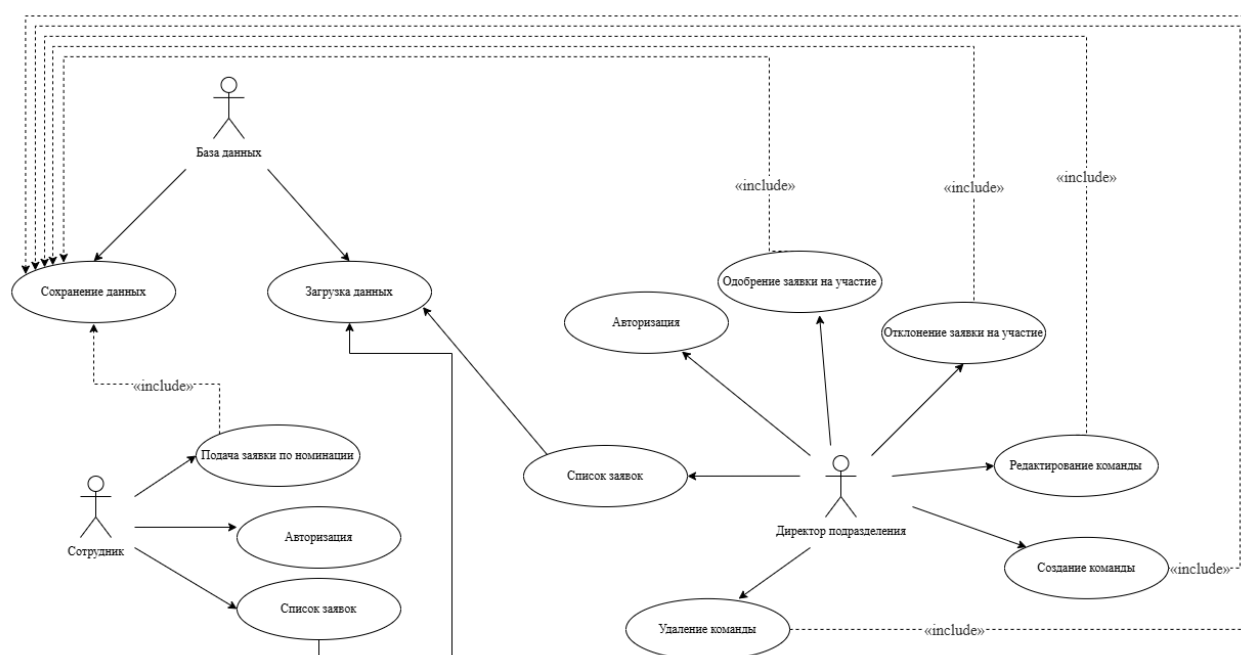


Рисунок 1 – Диаграмма вариантов использования этапа регистрации

Разрабатываемая автоматизированный модуль должен соответствовать следующим техническим требованиям:

- удобный и интуитивно понятный веб-интерфейс для работы с платформой;
- расположение элементов управления и информации должно обеспечивать пользователю быстрое ориентирование и эффективное выполнение задач;

– адаптивный дизайн для работы на различных устройствах (ПК, планшеты) и разрешениях экрана.

Диаграмма вариантов использования модуля на этапе проведения оценки представлен на рисунке 2.

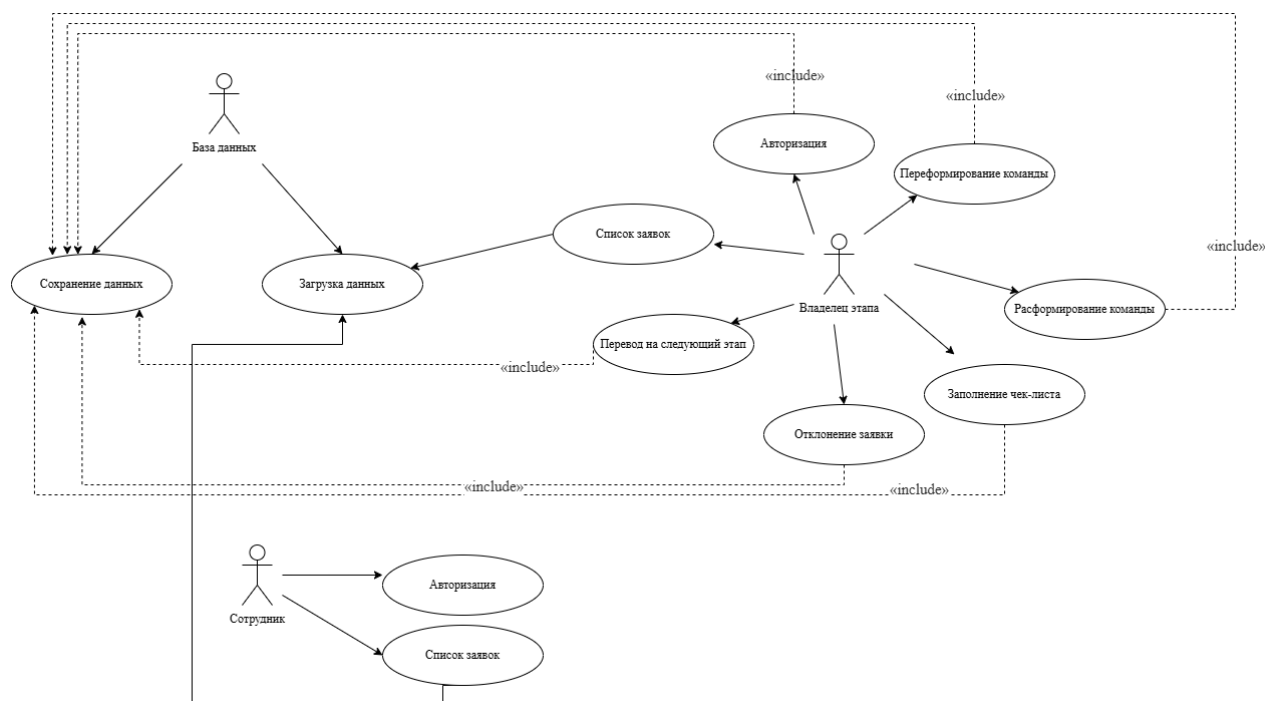


Рисунок 2 – Диаграмма вариантов использования этапа проведения оценки

Рабочие станции должны иметь следующие минимальные технические характеристики:

- операционная система Windows 7 и выше;
- процессор Intel Core 2,7 ГГц и выше;
- свободное место на жестком диске 50 ГБ и выше;
- оперативную память 8 ГБ и выше;
- монитор с диагональю не менее 16 дюймов;
- клавиатура;
- мышь.

Сервер в сети должен иметь следующие технические характеристики:

- операционная система Windows Server 2012 и выше;
- процессор Intel Xeon 3 ГГц и выше;

- свободное место на жестком диске 100 ГБ и выше;
- оперативную память 16 ГБ и выше;
- монитор с диагональю не менее 16 дюймов;
- клавиатура;
- мышь.

1.2 Выбор технологий и средств разработки приложения

В данном разделе будут описаны технологии и средства разработки, использованные при разработке программы по автоматизации работы общественного питания.

Ниже приведен список выбранных технологий и средств разработки:

- JS (React);
- HTML/CSS;
- Websoft HCM;
- MS SQL Server;
- VS Code;
- JSON.

1.2.1 Платформа Websoft HCM

Websoft HCM – это корпоративная платформа для комплексного управления персоналом, разработанная компанией Websoft. Платформа предоставляет большой набор инструментов и готовых модулей для автоматизации ключевых HR-процессов: адаптация, оценка компетенций, обучение, развитие сотрудников.

Платформа специализирована на автоматизации процессов управления персоналом, предоставляя готовые решения для оценки компетенций, управления заявками и документооборотом, что значительно ускорит разработку модуля по сравнению с созданием аналогичного функционала с нуля.

Websoft HCM позволяет разрабатывать серверную бизнес-логику на

					ПГУ 1.090501.12.001 ПЗ	Лист
						15
Изм.	Лист	№ докум.	Подпись	Дата		

языке JavaScript и C#, обеспечивая единообразие технологического стека с клиентской частью и упростит взаимодействие между разработчиками. Платформа изначально спроектирована для бесшовной интеграции с внутренними системами компании, что критично для автоматизации процессов, затрагивающих различные подразделения и существующие данные о сотрудниках.

Таким образом, использование Websoft HCM в качестве серверной платформы позволяет эффективно реализовать сложную бизнес-логику модуля оценки компетенций, обеспечить его интеграцию в существующую ИТ-инфраструктуру компании и сократить сроки разработки за счет использования специализированных HR-инструментов.

1.2.2 Выбор языка программирования

Для разработки автоматизированного модуля, предназначенного для многоэтапной оценки сотрудников, необходимо выбрать язык программирования, который обеспечит эффективную реализацию как клиентской, так и серверной части приложения. Рассмотрены несколько популярных решений: JavaScript, Python, Java и PHP.

JavaScript – универсальный язык, который может использоваться как для клиентской, так и для серверной разработки. На стороне клиента он позволяет достигнуть динамики взаимодействия с интерфейсом, без использования перезагрузки страницы [10]. В платформа Websoft HCM, JavaScript применяется в виде server-side кода, что упрощает разработку за счет использования единого языка на всех уровнях приложения.

PHP – распространённый язык программирования общего назначения с открытым исходным кодом, который сконструировали специально для веб-разработки и который встраивают в HTML-разметку [4]. Хотя язык широко используется в веб-разработке, но архитектурно не совместим с используемой платформой, что делает его применение нецелесообразным в рамках данного проекта.

Python – это простой в освоении и мощный язык программирования. Он

обладает эффективными высокоуровневыми структурами данных и простым, но эффективным подходом к объектно-ориентированному программированию [3]. Несмотря на простоту синтаксиса и удобство при создании прототипов, у языка отсутствует высокая производительность при работе с большим числом одновременных запросов.

В результате анализа основным языком проекта стал JavaScript, который продемонстрировал максимальную адаптивность под выбранную архитектуру. Его применение оказалось наиболее рациональным по следующим причинам:

На клиентской стороне JavaScript используется совместно с библиотекой React, что обеспечивает быстрое и интерактивное взаимодействие с интерфейсом, включая моментальное обновление данных.

На сервере язык применяется в рамках платформы Websoft HCM, что позволяет реализовать бизнес-логику, интеграцию с корпоративными системами и управление данными в едином технологическом пространстве.

Принятие решения в пользу использования одного языка программирования на всех уровнях системы позволило добиться важного преимущества – унификации процесса разработки. Это снизило вероятность ошибок при интеграции, упростило сопровождение кода и повысило общую производительность команды. Таким образом, JavaScript оказался наиболее подходящим выбором с точки зрения технологической целесообразности, скорости реализации и последующей поддержки.

1.2.3 Интерфейс и визуализация

Фронтенд приложения построен с использованием HTML и CSS, где HTML отвечает за структуру документа, а CSS – за стили и адаптацию интерфейса. Для стилизации применена библиотека styled-components, которая позволяет управлять стилями прямо в JavaScript, что обеспечивает удобство и модульность.

Управление состоянием приложения реализуется с использованием библиотеки effector-react, она позволяет централизовать хранение данных, эффективно обновлять интерфейс в ответ на действия пользователя и

исключить лишние перерисовки компонентов.

Осново полагающим элементом клиентской части стал фреймворк React. Он предоставляет компонентный подход, благодаря которому интерфейс разбивается на независимые блоки, упрощающие повторное использование и тестирование. Получение данных происходит через библиотеку Axios, позволяющую организовать асинхронное взаимодействие с сервером.

Для обеспечения адаптивности интерфейса применяются CSS Grid, Flexbox и медиа-запросы. Это позволяет подстраивать внешний вид приложения под экраны различных устройств.

Отказ от типовых CSS-фреймворков вроде Bootstrap обусловлен стремлением к полной кастомизации интерфейса и оптимизации производительности. Такой подход обеспечивает лёгкость интерфейса, минимализм и высокую скорость отклика.

1.2.4 База данных и работа с данными

Современные веб-приложения использование реляционных баз данных – неотъемлемый инструмент для структурирования данных. Таблицы связаны между собой и содержат уникальные объекты, а столбцы – его атрибуты. Для модуля требуется надежность и производительность, поэтому в качестве СУБД был выбран Microsoft SQL Server.

MS SQL Server – это одна из самых популярных систем управления базами данных (СУБД), разработанная корпорацией Microsoft. Она используется для хранения, обработки и анализа информации в различных сферах – от веб-приложений до корпоративных информационных систем. Однако, перед тем как выбрать SQL Server, важно понять его особенности, преимущества и недостатки, а также определить, подходит ли он для конкретных задач [9].

Главные особенности, которые повлияли на выбор MS SQL Server:

– полная интеграция с системой Microsoft, которая дает возможность безпроблемное использование с платформой Websoft HCM, а также дает бесшовную интеграцию Windows-серверами;

- обеспечивает безопасность данных, используя алгоритмы шифрования, аудита;
- имеет высокую оптимизацию и кэширование запросов;
- позволяет работать с JSON и XML, что упрощает общение с клиентской частью.

Выбор MS SQL Server обеспечит стабильную работу модуля, возможность глубокой аналитики и долгосрочное хранение данных. Это делает систему гибкой и готовой к масштабированию в будущем.

1.2.5 Среда разработки

Visual Studio Code (VS Code) – это бесплатный и очень легковесный редактор кода, разработанный компанией Microsoft. Обладает широким выбором средств, для веб-разработки.

VS Code обладает множеством преимуществ, которые выделяют его на фоне других сред разработки. Поддержка JavaScript и библиотеки React реализована превосходной дает пользователю: подсветку синтаксиса, интеллектуальное автодополнение, навигацию по коду, отладка. Большое возможное количество расширений, способствует к гибкой настройке среды. Встроенный терминал, позволяет вызывать нужные команды, не сворачивая среду.

Глубокая встроенная поддержка системы контроля версий git предоставляет удобный интерфейс для управления изменениями. VS Code отличается легковесностью и производительностью, что позволяет использование на большом количестве систем.

1.2.6 Обмен данными JSON

JSON (JavaScript Object Notation) – стандартный, легковесный текстовый формат, который предназначен для обмена данными, основанный на синтаксисе объектов JavaScript. Хотя произошел формат из JavaScript, но JSON является самостоятельным и широко поддерживаемым большинством современных платформ и языков программирования.

Выбор JSON в качестве основного формата передачи данных между

клиентом и сервером модуля, обусловлен его значительным преимуществом такими как: простота синтаксиса, минимальный размер передачи данных, идеально интегрируется с JavaScript, идеально позволяет представить сложные иерархические структуры.

1.3 Проектирование информационной системы

1.3.1 Паттерн RPC

RPC (Remote Procedure Call) – это метод организации межпроцессного взаимодействия, при котором клиентская программа может вызывать процедуры или функции на удалённом сервере [2] так, будто они выполняются локально. Данный подход обеспечивает абстракцию сетевой коммуникации за счёт инкапсуляции деталей передачи данных.

В представленной реализации RPC используется клиент-серверная модель с единым HTTP-эндпоинтом для всех операций. Это устраняет необходимость сложной маршрутизации множества URL, характерной для REST-архитектуры, и централизует управление запросами. Все запросы передаются исключительно через POST-метод, гарантируя единообразие формата передачи и обработки данных.

Каждый клиентский запрос содержит три обязательных поля:

- class - указывает целевой модуль или сервис на сервере;
- action - определяет конкретную операцию (метод), которую требуется выполнить;
- data - включает параметры вызова в структурированном виде (JSON).

Такая стандартизация упрощает парсинг запросов, валидацию и дальнейшую диспетчеризацию на стороне сервера.

Логика обработки реализуется следующим образом: сервер принимает запрос, идентифицирует целевой класс и метод, после чего выполняет соответствующую процедуру. Результат работы возвращается клиенту в виде ответа. Схема взаимодействия компонентов показана на Рисунке 3.

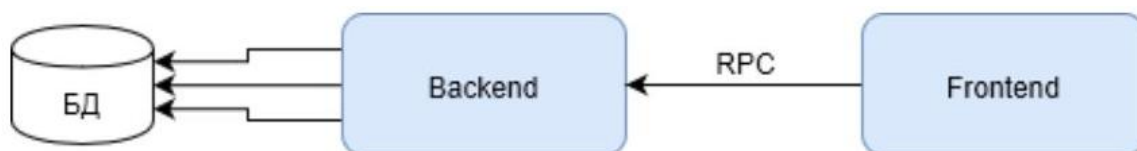


Рисунок 3 – Схема RPC-коммуникации клиент-сервер

В отличие от REST, который опирается на ресурсоориентированную модель, RPC фокусируется на выполнении действий. Это позволяет гибко описывать сложную бизнес-логику без привязки к концепции ресурсов.

В качестве основы коммуникации выбран RPC-подход, поскольку он обеспечивает:

Снижение сложности маршрутизации за счёт единой точки входа;

Упрощение разработки API благодаря стандартизированному формату запросов;

Эффективную обработку составных операций, требующих выполнения нескольких методов за один вызов;

Прямое соответствие между клиентскими вызовами и серверной логикой, что ускоряет интеграцию.

Данные преимущества особенно значимы для приложений с интенсивным обменом данными и разнообразными сценариями взаимодействия между клиентом и сервером.

1.3.2 Архитектура модуля

На рисунке 4 представлена архитектура модуля для автоматизации многоэтапной оценки сотрудников.

Взаимодействуя с окнами, формами или кнопками, являются частью компонента представления (View), предоставляя пользователю интерфейс для ввода данных и выполнения действий. Когда пользователь инициирует действия, нажимает на кнопки или наводит курсор, управление передается контроллеру (Controller). Основной задачей контроллера является обработка действий, полученных от компонента представления и валидация их. Если данные успешно проходят валидацию, то контроллер передает их компоненту

модели (Model), которая реализует бизнес-логику приложения и отвечает за обработку данных согласно заложенному в нее алгоритму.

Результаты обработки данных должны быть доведены до пользователя в компонент представления. Если все прошло успешно и компонент модели обработал данные, то пользователю приходит сообщение об успехе в соответствующий компонент представления, если в процессе валидации в компоненте контроллера произошла ошибка во введенных данных, то он напрямую информирует об это связанный с действием компонент представления. В свою очередь компонент представления обновляет интерфейс и отображает пользователю информационное сообщение.

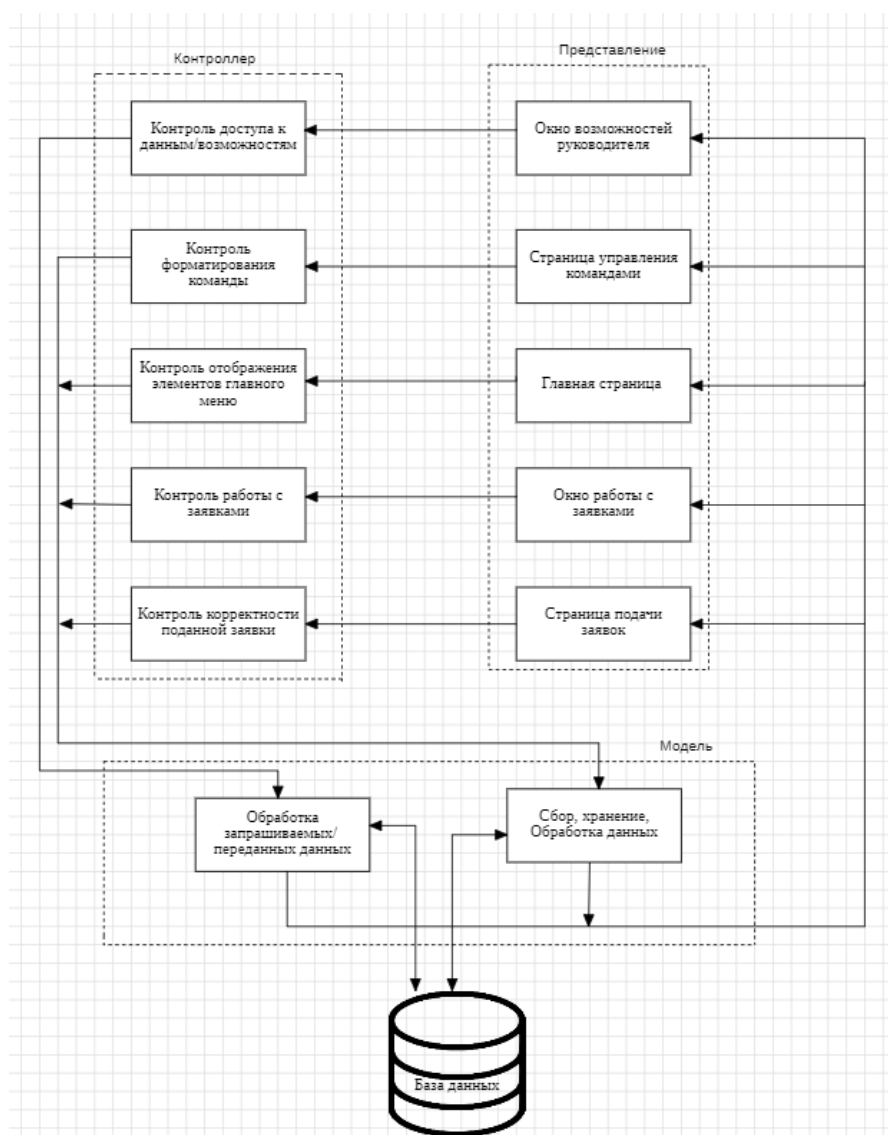


Рисунок 4 – Архитектура модуля

На рисунке 5 представлено взаимодействие компонентов приложения в виде диаграммы последовательности.

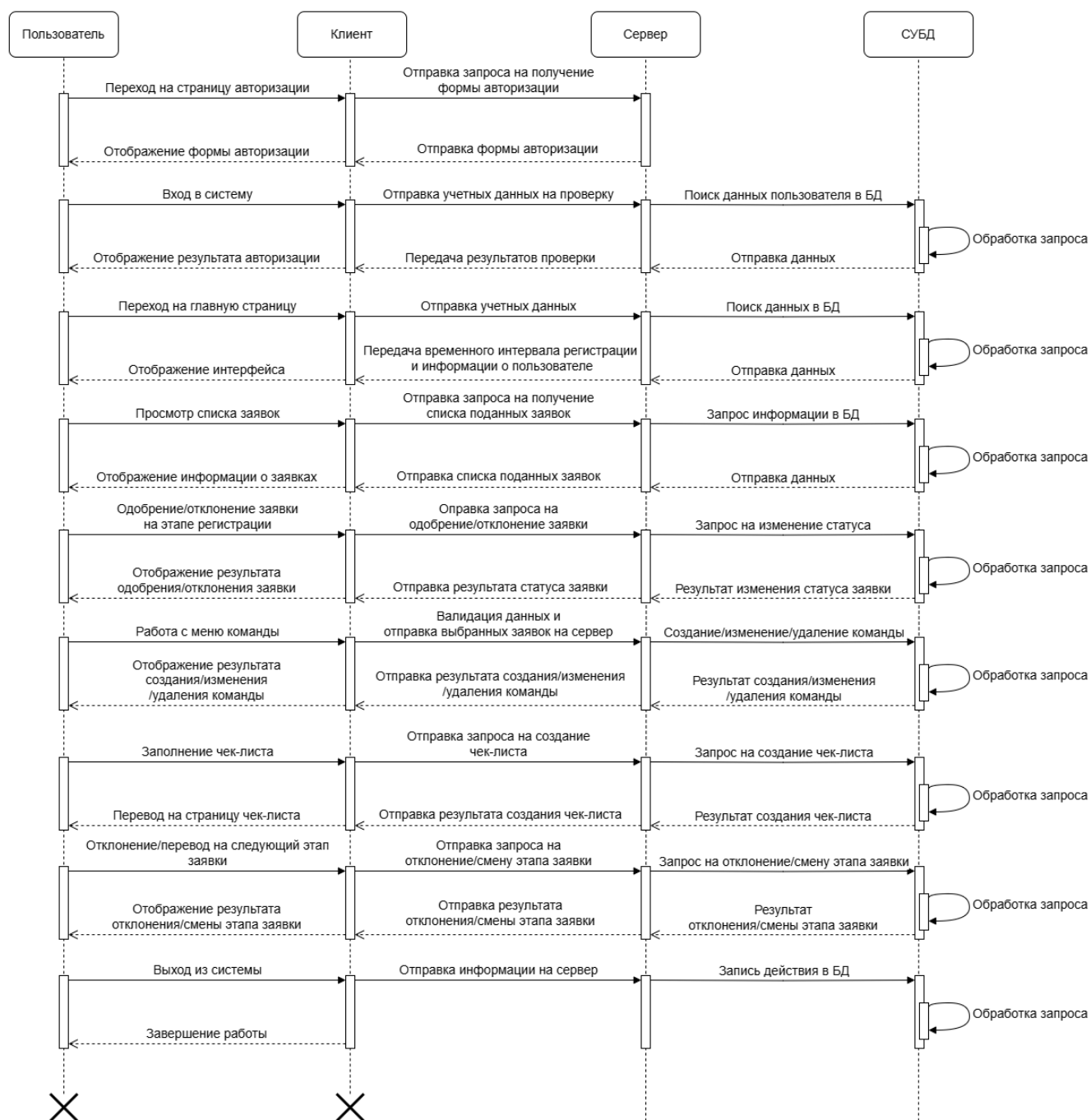


Рисунок 5 – Диаграмма последовательности

Вся работа с приложением будет осуществляться в виде диалога с пользователем. Диалоговая система состоит из трех ключевых компонентов (рисунок 6):

— клиентская часть (браузер), которая обеспечивает взаимодействие с

пользователем;

- серверная часть, которая обрабатывает запросы, отправленные пользователем, выполняет необходимые алгоритмы и возвращает запрашиваемые данные;

- база данных, в которой хранятся данные с учётом определённых правил и принципов управления.

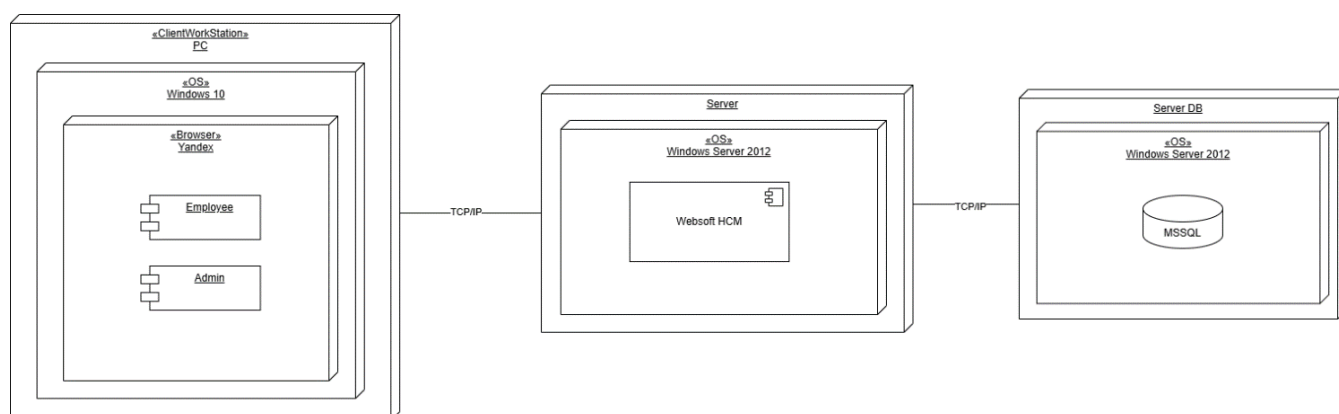


Рисунок 6 – Диаграмма развертывания приложения

1.3.3 Проектирование базы данных

Для хранения информации необходимо реализовать базу данных со следующими таблицами:

- participation – таблица, содержащая данные по заявкам, связанная с таблицами period, nomination, request (таблица 1.3.1);

- period – таблица, хранящая год проведения оценки и временной интервал этапов (таблица 1.3.2);

- nomination – таблица, хранящая номинации (таблица 1.3.3);

- event_appr – таблица, создающая связь между таблицами request и assessment_appraise (таблица 1.3.4);

- assessment_appraise – таблица, хранящая информацию о процедуре оценки (таблица 1.3.5);

- request – таблица, содержащая данные о заявке (таблица 1.3.6);

- collaborator – таблица, хранящая данные учетных записей сотрудников

(таблица 1.3.7);

– workflow – таблица, хранящая данные о процедуре документооборота

(таблица 1.3.8);

– aggregation_notif – таблица, содержащая агрегированные уведомления

(таблица 1.3.9);

– position – таблица, содержащая должности сотрудников (таблица 1.3.10);

– subdivision – таблица, содержащая подразделения сотрудников (таблица 1.3.11);

– ra – таблица, содержащая информацию анкеты процедуры оценки (таблица 1.3.12);

Таблица 1.3.1 – Атрибуты таблицы participation

Имя столбца	Тип данных	Описание	Роль атрибута
id	BIGINT	Уникальный идентификатор	РК
code	NVARCHAR(255)	Код участия	
name	NVARCHAR(255)	Наименование участника	
request_id	BIGINT	ID заявки	FK
period_id	BIGINT	ID периода	FK
nomination_id	BIGINT	ID номинации	FK
team_name	NVARCHAR(255)	Наименование команды	
region_manager_id	BIGINT	ID директор подразделения	FK
area_manager_id	BIGINT	ID районного менеджера	FK

Таблица 1.3.2 – Атрибуты таблицы period

Имя столбца	Тип данных	Описание	Роль атрибута
id	BIGINT	Уникальный идентификатор	РК
year	INT	Год	
request_start_date	DATE	Дата начала подачи заявок	
request_finish_date	DATE	Дата окончания подачи заявок	
stage_1_start_date	DATE	Дата начала этапа 1	
stage_1_finish_date	DATE	Дата окончания этапа 1	

Имя столбца	Тип данных	Описание	Роль атрибута
stage_2_start_date	DATE	Дата начала этапа 2	
stage_2_finish_date	DATE	Дата окончания этапа 2	
stage_3_start_date	DATE	Дата начала этапа 3	
stage_3_finish_date	DATE	Дата окончания этапа 3	
stage_4_start_date	DATE	Дата начала этапа 4	
stage_4_finish_date	DATE	Дата окончания этапа 4	

Таблица 1.3.3 – Атрибуты таблицы nomination

Имя столбца	Тип данных	Описание	Роль атрибута
id	BIGINT	Уникальный идентификатор	РК
code	NVARCHAR(255)	Код номинации	
name	NVARCHAR(255)	Наименование номинации	

Таблица 1.3.4 – Атрибуты таблицы event_appr

Имя столбца	Тип данных	Описание	Роль атрибута
id	BIGINT	Уникальный идентификатор	РК
name	NVARCHAR(255)	Наименование оценки	
request_id	BIGINT	ID заявки	FK
workflow_state	NVARCHAR(255)	Состояние ДО	
assessment_appraise_id	BIGINT	ID оценки	FK

Таблица 1.3.5 – Атрибуты таблицы assessment_appraise

Имя столбца	Тип данных	Описание	Роль атрибута
id	BIGINT	Уникальный идентификатор	РК
code	NVARCHAR(255)	Код оценки	
name	NVARCHAR(255)	Наименование оценки	
status	INT	Статус	
is_model	BIT	Флаг эталонной процедуры	
start_date	DATETIME	Дата начала	
end_date	DATETIME	Дата окончания	
person_id	BIGINT	ID сотрудника	FK
workflow_id	BIGINT	ID ДО	FK

Таблица 1.3.6 – Атрибуты таблицы request

Имя столбца	Тип данных	Описание	Роль атрибута
id	BIGINT	Уникальный идентификатор	РК
code	NVARCHAR(255)	Код заявки	
type	NVARCHAR(255)	Тип заявки	
person_id	BIGINT	ID инициатора заявки	FK
is_group	BIT	Групповая заявка	
object_id	BIGINT	ID сотрудника	FK
workflow_id	BIGINT	ID ДО	FK
nomination_id	BIGINT	ID номинации	FK

Таблица 1.3.7 – Атрибуты таблицы collaborator

Имя столбца	Тип данных	Описание	Роль атрибута
id	BIGINT	Уникальный идентификатор	РК
code	NVARCHAR(255)	Код сотрудника	
fullname	NVARCHAR(255)	Полное имя	
login	NVARCHAR(255)	Логин	
email	NVARCHAR(255)	Электронная почта	
phone	NVARCHAR(255)	Телефон	
birth_date	DATE	Дата рождения	
sex	NVARCHAR(255)	Пол	
pict_url	NVARCHAR(255)	URL фотографии	
position_id	BIGINT	ID должности	FK
position_parent_id	BIGINT	ID подразделения	FK
is_dismiss	BIT	Уволен (да/нет)	
position_date	DATE	Дата назначения на должность	
hire_date	DATE	Дата приема на работу	
dismiss_date	DATE	Дата увольнения	

Таблица 1.3.8 – Атрибуты таблицы workflow

Имя столбца	Тип данных	Описание	Роль атрибута
id	BIGINT	Уникальный идентификатор	РК
code	NVARCHAR(255)	Код	
name	NVARCHAR(255)	Наименование	
condition_eval_str	NVARCHAR(255)	Условия выполнения	

Таблица 1.3.9 – Атрибуты таблицы aggregation_notif

Имя столбца	Тип данных	Описание	Роль атрибута
id	BIGINT	Уникальный идентификатор	РК
code	NVARCHAR(255)	Код уведомления	

Имя столбца	Тип данных	Описание	Роль атрибута
recipient_id	BIGINT	ID получателя	FK
object_id	BIGINT	ID отправителя	FK
is_send	BIT	Отправлено (да/нет)	

Таблица 1.3.10 – Атрибуты таблицы position

Имя столбца	Тип данных	Описание	Роль атрибута
id	BIGINT	Уникальный идентификатор	PK
name	NVARCHAR(255)	Наименование должности	
parent_object_id	BIGINT	ID подразделения	FK
basic_collaborator_id	BIGINT	ID сотрудника	FK
is_boss	BIT	Руководитель (да/нет)	
position_date	DATE	Дата назначения в должности	
position_finish_date	DATE	Дата завершения работы в должности	
is_position_finished	BIT	Действие должности прекращено	

Таблица 1.3.11 – Атрибуты таблицы subdivision

Имя столбца	Тип данных	Описание	Роль атрибута
id	BIGINT	Уникальный идентификатор	PK
name	NVARCHAR(255)	Наименование подразделения	
parent_object_id	BIGINT	ID родительского подразделения	FK
is_disbanded	BIT	Расформировано (да/нет)	

Таблица 1.3.12 – Атрибуты таблицы ra

Имя столбца	Тип данных	Описание	Роль атрибута
id	BIGINT	Уникальный идентификатор	PK
name	NVARCHAR(255)	Наименование	
assessment_appraise_id	BIGINT	ID оценки	FK
person_id	BIGINT	ID сотрудника	FK
expert_person_id	BIGINT	ID эксперта	FK
period_start	DATETIME	Дата начала оценки	
period_end	DATETIME	Дата окончания оценки	
is_done	BIT	Завершено (да/нет)	
status	NVARCHAR(255)	Статус	
overall	FLOAT(53)	Общий балл	
workflow_id	BIGINT	ID ДО	FK

Физическая модель базы данных приведена на рисунке 7.

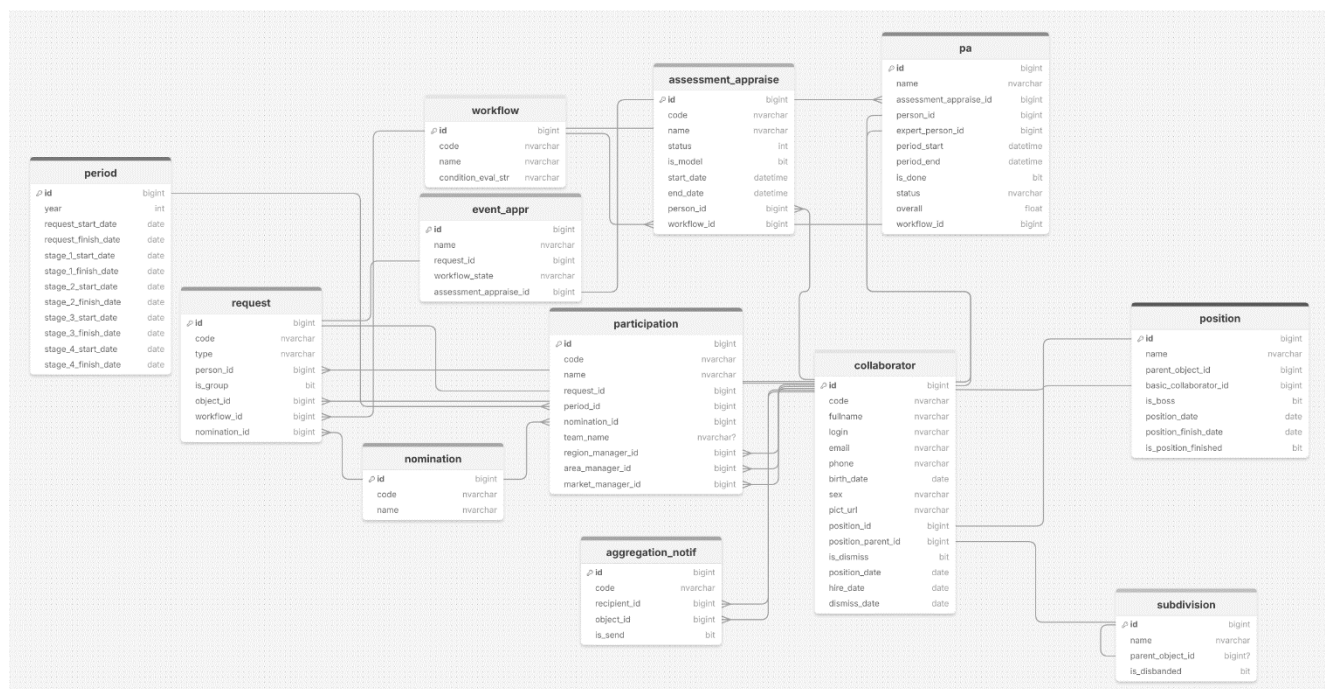


Рисунок 7 – Физическая модель БД

1.4 Разработка приложения

1.4.1 Хранение данных

Одной из ключевых архитектурных особенностей Websoft HCM является использование гибридной модели хранения данных. Система предоставляет возможность хранить информацию в виде структурированных XML-документов. Этот формат очень удобен при хранении сложных иерархических и постоянно изменяемых данных. Однако Websoft HCM не останавливается на простом хранении документа в виде XML структуры. Система автоматически и прозрачно связывает каждый XML-документ с реляционной SQL таблицей. Такая интеграция позволяет взаимодействовать с данными через мощный и универсальный SQL-интерфейс.

Процесс расширения функционала и создания собственных таблиц реализован в Websoft HCM с использованием концепции «Тип документа». При необходимости хранить новый вид информации, уникальный для конкретного

предприятия, создается новый «Тип документа» (рисунок 8).

Участие в многоэтапной оценке — Тип документа

Код: ss_participation, Название: Участие в многоэтапной оценке, Категория:

Имя объекта: ss_participation

Поле в качестве заголовка: Наименование(name)

Поле: code, Название: Код, Страница:

Тип поля: Строка, ☒ Поле в каталоге, ☐ Индексированное поле, ☐ Массив

Поле: name, Название: Наименование, Страница:

Тип поля: Строка, ☒ Поле в каталоге, ☐ Индексированное поле, ☐ Массив

Рисунок 8 – Тип документа участия в многоэтапной оценке

1.4.2 Описание программы

Описание клиентской части.

Клиентская часть реализована по модульному принципу с использованием библиотеки React. Такая архитектура обеспечивает чёткое разделение функций, что способствует масштабируемости и удобству сопровождения системы.

В папке /src/libs/constants хранятся глобальные константы. Эти значения доступны во всех модулях приложения, что позволяет производить изменения из одного места.

```
export const BACK_URL = '/growth_skill/back/router.html';
export const REQUEST_TIMEOUT = 5 * 60 * 1000;
```

Модуль /src/libs/request настраивает HTTP-клиент. Место откуда происходит управление всеми запросами.

```
private const axiosInstance = axios.create({
  timeout: REQUEST_TIMEOUT,
});

axiosInstance.interceptors.request.use(
```

```

    (config) => config,
    (error) => {
      console.error('Request Error:', error);

      return Promise.reject(error);
    },
  );

  axiosInstance.interceptors.response.use(
    (response) => response,
    async (error) => {
      if (!error?.response) {
        console.error('Network Error:', error?.message);

        return Promise.reject(error);
      }

      const status = error?.response?.status;
      const data = error?.response?.data;

      switch (status) {
        case 401:
          console.error('NoAuthorization:', data);
          break;
        case 403:
          console.error('Forbidden:', data);
          break;
        default:
          console.error(`HTTP Error ${status}:`, data);
      }
      return Promise.reject(error);
    },
  );
);

```

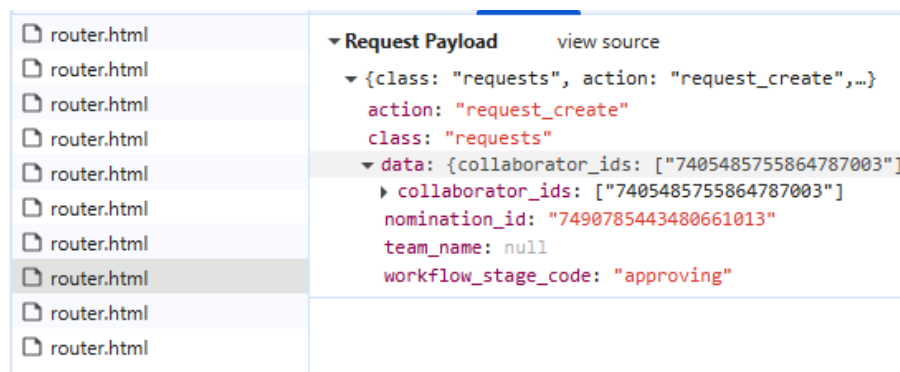


Рисунок 6 – Модель клиентского запроса

Каждый API-модуль используется логику взаимодействия с определённым серверным endpoint, используя RPC-паттерн (рисунок 6). Ниже приведен пример кода из файла request.js.

```
const requestCreate = (nomination, userId) =>
  axiosInstance.post(BACK_URL, {
    class: 'requests',
    action: 'create',
    data: { nomination_code: nomination, user_id: userId }
  });
```

Описание серверной части.

Файл роутинга router.html сначала инициализирует доступ пользователя через вызовы access_init.html для настройки прав доступа и tools_web.user_init для определения текущего пользователя. Из полученного запроса происходит извлечение значений полей class, action и data. Если class или action отсутствуют, клиент получает сообщение об ошибке.

На основе значения class формируется путь к JavaScript-файлу, содержащему необходимую бизнес-логику. При необходимости может быть сброшен кэш формы через DropFormsCache. После этого файл загружается как псевдо-документ с помощью OpenCodeLib, и нужный метод вызывается через CallObjectMethod, куда передаются data, текущий пользователь и объект запроса. Результат возвращается в формате JSON. Пример серверного кода представлен ниже.

```
Server.Execute(AppDirectoryPath() +
"/wt/web/include/access_init.html");

SessionUser = tools_web.user_init( Request, Request.Query );
Request.AddRespHeader("Content-Type", "application/json");

_options = Request.Body;
var stream = Request.RespStream;
try {
    _options = tools.read_object(_options);

    _class = _options.GetOptProperty("class", null);
    action = _options.GetOptProperty("action", null);
    data = _options.GetOptProperty("data", {});

    if(action==null || _class==null)
        stream.WriteStr('Ошибка нужно отправлять action
class');

    var libUrl = 'x-local://wt/web/growth_skill/back/' +
```

					ПГУ 1.090501.12.001 ПЗ	Лист
						32
Изм.	Лист	№ докум.	Подпись	Дата		

```

_class+'.js';

var apiDoc = OpenCodeLib(libUrl);

res = CallObjectMethod(apiDoc, action, [data,
curUser, Request]);
stream.WriteString(tools.object_to_text(res, "json"));

} catch(err) {
    alert('report_api', err);
    errorObj = {
        error: true,
        message: RValue(err)
    };
    stream.WriteString(tools.object_to_text(errorObj,
"json"));
}

```

Все параметры, поступающие со стороны клиента, проходят обязательную валидацию – при обнаружении некорректных данных пользователь получает соответствующее сообщение об ошибке.

Для удобства повторного использования и упрощения вызова функций системы была сформирована библиотека custom_skill_tools, включающая все необходимые методы для обработки заявок. Вызов нужных функций осуществляется через получение псевдо-документа и обращение к соответствующему методу этого объекта.

```

function request_create(data, curUser) {
    var returnObj = {
        data: null,
        error: false,
        message: "success",
    };

    try {
        if (!OptInt(curUser.id)) {
            return { data: [], error: true, message: "Ошибка: нет id
пользователя" };
        }

        var skill_tools = OpenCodeLib(
            FilePathToUrl( AppDirectoryPath() + "\\custom_tools
\\custom_skill_tools.js"
        )
    );

    var result = skill_tools.requestCreate(

```

```

        curUser.id,
        data.GetOptProperty("collaborator_ids", []),
        data.GetOptProperty("nomination_id", undefined),
        data.GetOptProperty("team_name", undefined),
        data.GetOptProperty("workflow_stage_code", undefined)
    );
    return returnObj;
} catch (error) {
    return { data: null, error: true, message: "Ошибка: " +
error };
}
}

```

Получение данных из базы осуществляется с помощью функции XQuery, которая поддерживает две формы записи запросов: в виде SQL (с обязательным указанием префикса «sql:») и на языке XQuery. Результаты запроса возвращаются в виде массива, и для получения первого элемента используется функция ArrayOptFirstElem. Она возвращает сам элемент, если он существует, или неопределенное значение, если массив пуст.

```

function requestCreate(resp_col_id, collaborator_ids,
nomination_id, team_name, workflow_stage_code) {
    iPersonID = OptInt(resp_col_id);
    nomination_id = OptInt(nomination_id);

    if (ArrayCount(collaborator_ids) > 1) {
        for (collaborator_id in collaborator_ids) {
            if (!isBossForCollaborator(collaborator_id,
resp_col_id)) {
                throw 'отказано в доступе';
            }
        }
    } else {
        if (OptInt(ArrayOptFirstElem(collaborator_ids)) !=
OptInt(resp_col_id, 0)) {
            throw 'отказано в доступе';
        }
    }

    getType = ArrayOptFirstElem(XQuery("for $elem in
request_types where $elem/code = 'league' return $elem"));
    if (getType == undefined) {
        throw "не удалось получить идентификатор типа заявки
по коду - league";
    }
}

```

Создание элементов таблицы осуществляется через получение экземпляра каталога с использованием стандартного метода библиотеки tools -

					ПГУ 1.090501.12.001 ПЗ	Лист
						34
Изм.	Лист	№ докум.	Подпись	Дата		

new_doc_by_name, в который передаётся имя нужного каталога. Привязка к базе данных выполняется вызовом метода BindToDB у полученного экземпляра. Для обращения к полям каталога необходимо перейти к параметру TopElem и далее обращаться к нужным полям таблицы. Значения этих полей можно изменять, а для сохранения внесённых изменений следует вызвать метод Save у экземпляра.

```
// Карточки участников
docSkillParticip = tools.new_doc_by_name('cc_participation',
false);
docSkillParticip.BindToDb();
docSkillParticip.TopElem.name = Year(Date()) + " - №" +
oResCreateRequest.doc_request.TopElem.code + " - " + club_name;
if (team_name) {
    docSkillParticip.TopElem.team_name = team_name;
}
docLeagueParticip.Save();
```

Функция tools.open_doc служит для загрузки документа по заданному идентификатору – она принимает на вход числовое значение и возвращает экземпляр псевдо-документа, если документа с указанным ID не существует, tools.open_doc вернёт undefined, что позволяет сразу проверить корректность передаваемого идентификатора.

```
function requestApprove(request_id) {
    request_id = OptInt(request_id);
    requestDoc = tools.open_doc(request_id);
    if (requestDoc == undefined) {
        throw "заявка с id [ " + request_id + " ] не
найдена";
    }

    if (nomination_code == "party" && !requestDoc.TopElem.
is_group) {
        requestDoc.TopElem.workflow_state = status_workflow
[ArrayCount(status_workflow) - 1];
        requestDoc.TopElem.status_id = "close";
    }
}
```

В данной подсистеме реализован ряд JavaScript-модулей, обеспечивающих взаимодействие с пользовательским интерфейсом и серверной частью. Ниже представлены основные файлы и реализуемые в них

методы.

Модуль dictionaries.js предназначен для работы со справочниками и имеет методы:

- get_nominations() – возвращает список всех номинаций;
- get_workflow_stages() – возвращает этапы документооборота для заявок;
- get_request_statuses() – возвращает возможные статусы заявок.

Модуль skill.js предназначен для получения информации о текущем периоде. Имеет 1 метод.

get_current_period(data) – возвращает данные текущего периода.

Модуль requests.js обеспечивает управление заявками и имеет методы:

- get_requests_by_user_id(data, curUser) – возвращает заявки, связанные с текущим пользователем;
- get_count_requests_by_cur_period() – возвращает количество заявок за текущий период;
- request_cancel(data, curUser) – отклоняет заявку, если пользователь обладает соответствующими правами;
- request_approve(data) – переводит заявку на следующий этап документооборота;
- request_create(data, curUser) – создаёт новую заявку;
- request_update(data, curUser) – обновляет состав участников или название команды в групповой заявке;
- get_requests(data, curUser) – возвращает список заявок, связанных с текущим пользователем;
- get_permission_rules(data, curUser) – возвращает права доступа пользователя на вкладке «Судейство»;
- create_checklist(data, curUser) – создаёт чек-лист оценки для заявки (индивидуальной или групповой).

Модуль session.js отвечает за получение информации о текущем пользователе. Имеет 1 метод.

get_user_info(data, curUser) – возвращает информацию о текущем

пользователе.

Модуль `teames_manage.js` реализует логику формирования команд. Имеет 1 метод.

`get_available_collaborators_for_team(data, curUser)` – возвращает список сотрудников, которых можно добавить в команду.

Содержимое методов приведены в приложении Б.

1.4.3 Руководство пользователя

Пользовательский интерфейс должен быть удобен, прост и интуитивно понятен каждому пользователю, вне зависимости от уровня его подготовки. На рисунке 6 представлена карта навигации по страницам приложения.

Ниже представлено подробное описание интерфейса каждой страницы.

Элементы интерфейса страницы «Главная страница» представлены на рисунке 8:

- кнопка для перехода на главную страницу (1);
- кнопка для перехода на страницу новостей (2);
- кнопка для перехода на страницу судейства (3);
- кнопка для перехода на страницу моих заявок (4);
- кнопка для подачи заявки (5);
- выпадающий список с выбором выхода (6).

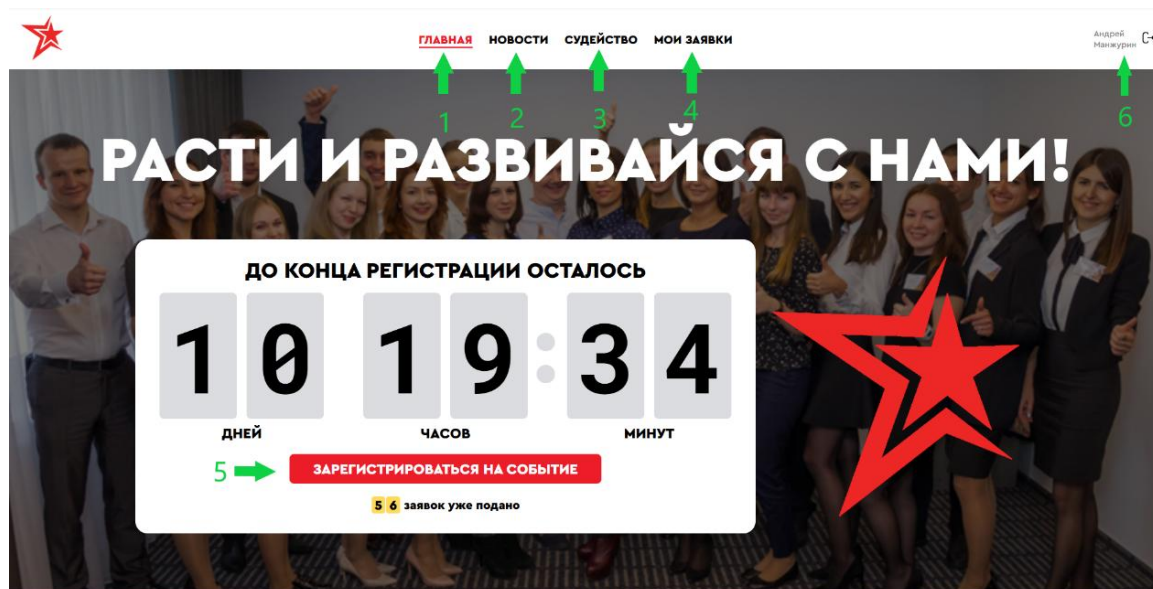


Рисунок 8 – Главная страница

Элементы интерфейса модального окна главной страницы при нажатии на кнопку «Зарегистрироваться на событие» представлены на рисунке 9:

- подтверждение выбора, подача заявки по выбранной номинации (1);
- закрытие модального окна (2);
- список с возможностью выбора номинации (3).

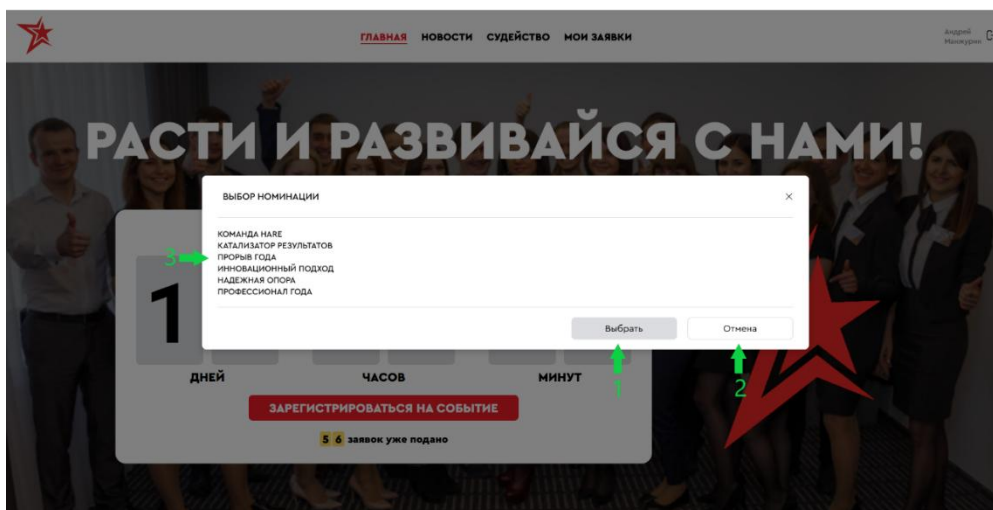


Рисунок 9 – Подача заявки на номинацию

Элементы интерфейса выпадающий список главной страницы при наведении на ФИО сотрудника представлены на рисунке 10:

- выход из учетной записи (1);
- переход на основной портал системы (2);
- выход из учетной записи (3).

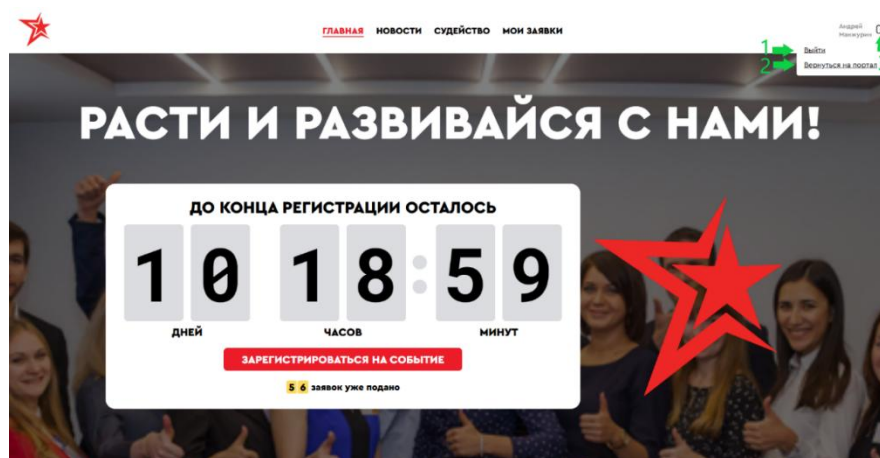


Рисунок 10 – Выпадающий список с выходом со страницы

Элемент интерфейса страницы «Судейство» представлены на рисунке 11.

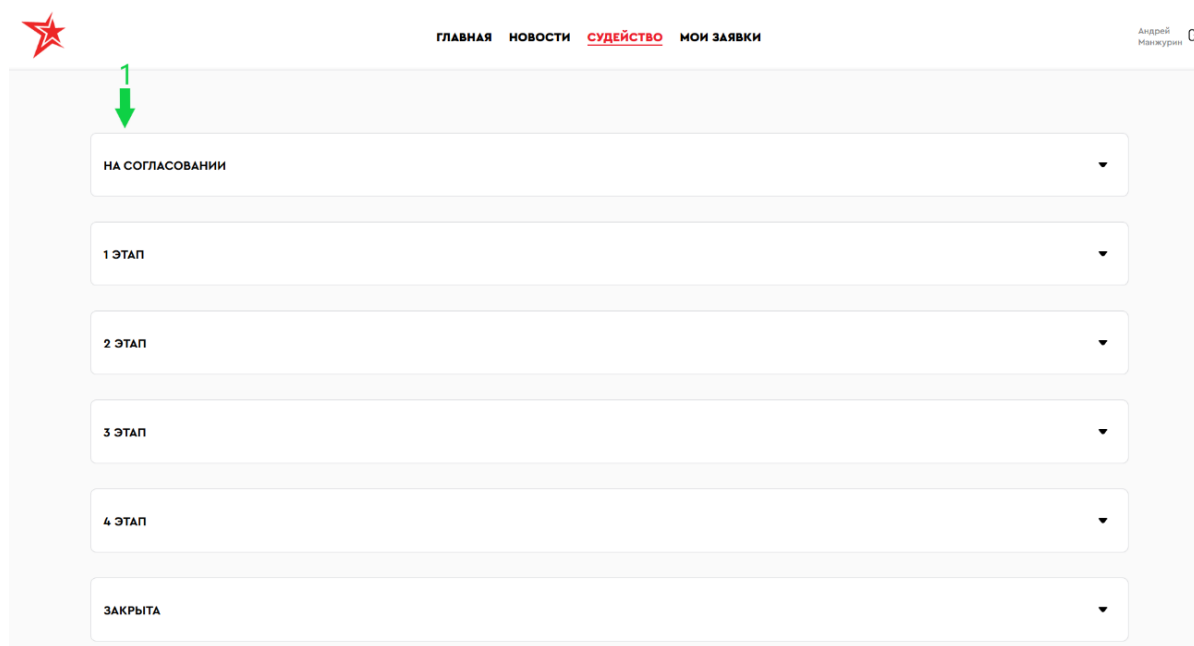


Рисунок 11 – Страница Судейство с закрытыми вкладками

Элементы интерфейса страницы «Судейство» для вкладки согласования командных номинаций представлены на рисунке 12:

- кнопка с выбором номинации (1);
- кнопка для перехода на страницу управление командой (2).

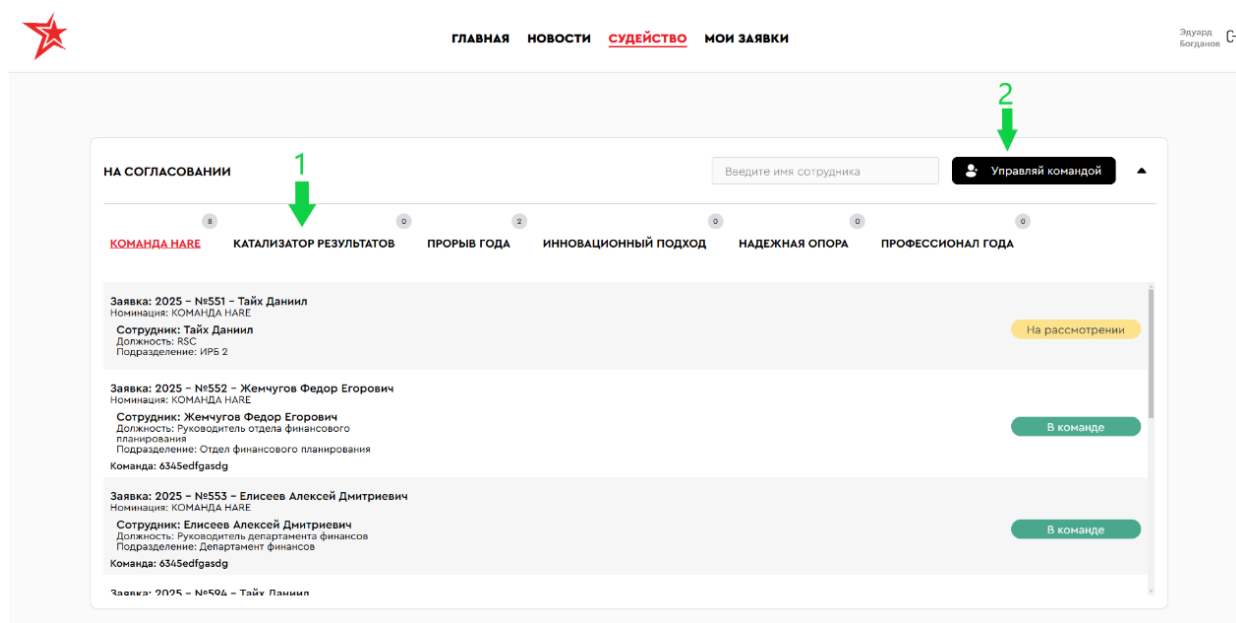


Рисунок 12 – Вкладка согласования командной номинации

Элементы интерфейса страницы «Судейство» для вкладки согласования индивидуальных номинаций представлены на рисунке 13:

- кнопка одобрения заявки (1);
- кнопка отклонения заявки (2).

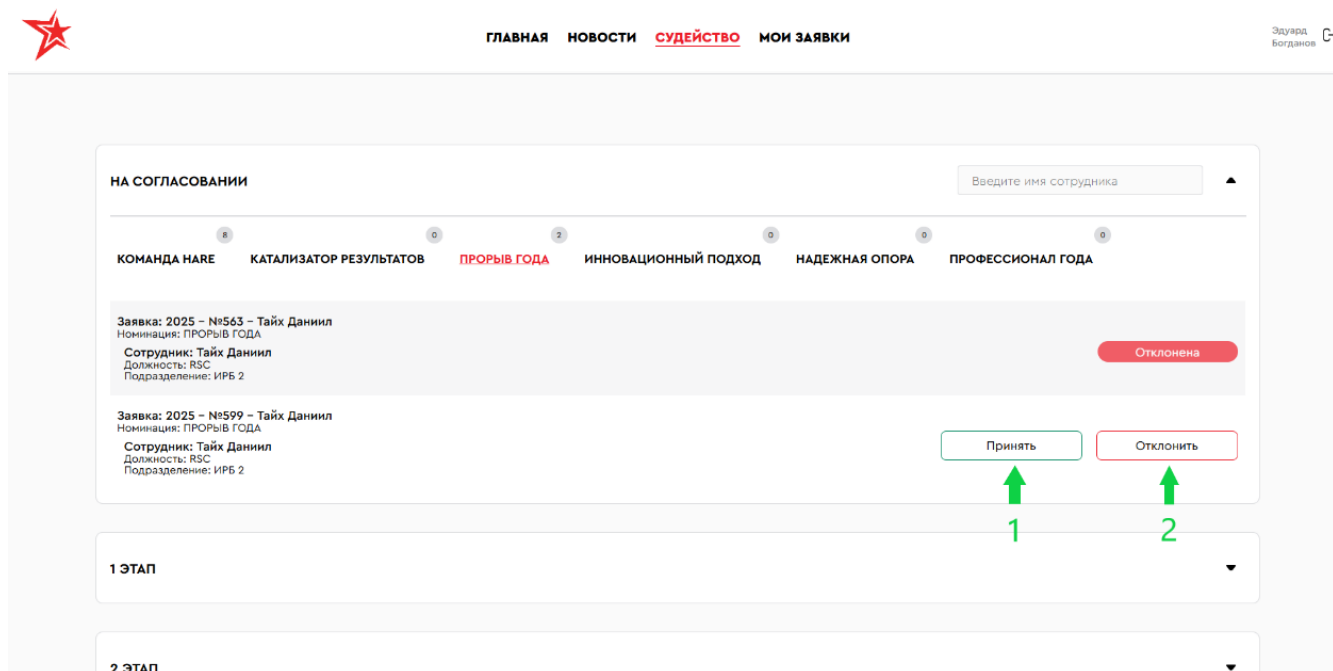


Рисунок 13 – Вкладка согласования индивидуальной номинации

Элементы интерфейса страницы «Судейство» для вкладки 1 этапа индивидуальных номинаций представлены на рисунке 14:

- кнопка заполнения чек-листа (1);
- кнопка перевода заявки на следующий этап (2);
- кнопка отклонения заявки (3);
- поиск по ФИО сотрудника (4);
- кнопка перехода в заполненный чек-лист (5).

НА СОГЛАСОВАНИИ

Введите имя сотрудника

КОМАНДА НАРЕ КАТАЛИЗАТОР РЕЗУЛЬТАТОВ ПРОРЫВ ГОДА ИННОВАЦИОННЫЙ ПОДХОД НАДЕЖНАЯ ОПОРА ПРОФЕССИОНАЛ ГОДА

Список заявок пуст

1 ЭТАП

Введите имя сотрудника

КОМАНДА НАРЕ КАТАЛИЗАТОР РЕЗУЛЬТАТОВ ПРОРЫВ ГОДА ИННОВАЦИОННЫЙ ПОДХОД НАДЕЖНАЯ ОПОРА ПРОФЕССИОНАЛ ГОДА

Тайх Даниил
Ресторан ИРБ 2

Манжурин Андрей Владимирович
Corp Dep 1
Общая оценка: 2.20

Эдуард Богданов
20.05.2025

Балл: 2.2

Посмотреть чек-лист

Заполнить чек-лист

На следующий этап

Отклонить

Рисунок 14 – Вкладка 1 этапа индивидуальной номинации

Элементы интерфейса страницы «Управление командой» представлены на рисунке 15:

- кнопка создания команды (1);
- кнопка удаления команды (2);
- кнопка редактирования команды (3);
- кнопка возвращения на вкладку судейство (4).

★

ГЛАВНАЯ НОВОСТИ СУДЕЙСТВО МОИ ЗАЯВКИ

Эдуард Богданов

Назад

ТЕКУЩИЕ КОМАНДЫ

Создать команду

Команда: Только вперед
Номинация: КОМАНДА НАРЕ

Сотрудник: Домнин Степан Владимирович
Должность: Test
Подразделение: Support Center

Сотрудник: Елисеев Алексей Дмитриевич
Должность: Руководитель департамента финансов
Подразделение: Департамент финансов

Сотрудник: Елистратов Максим
Должность: IDолжность Requests Test 6
Подразделение: Corp Dep 1

Сотрудник: Женчугов Федор Егорович
Должность: Руководитель отдела финансового планирования
Подразделение: Отдел финансового планирования

Рисунок 15 – Страница управление командой

Элементы интерфейса модального окна страница управления командой при нажатии на кнопку «Создать команду» представлены на рисунке 16:

- наименование команды (1);
- выбор заявок по командной номинации, включаемых в заявку (2);
- кнопка создания команды (3);
- кнопка закрытия модального окна.

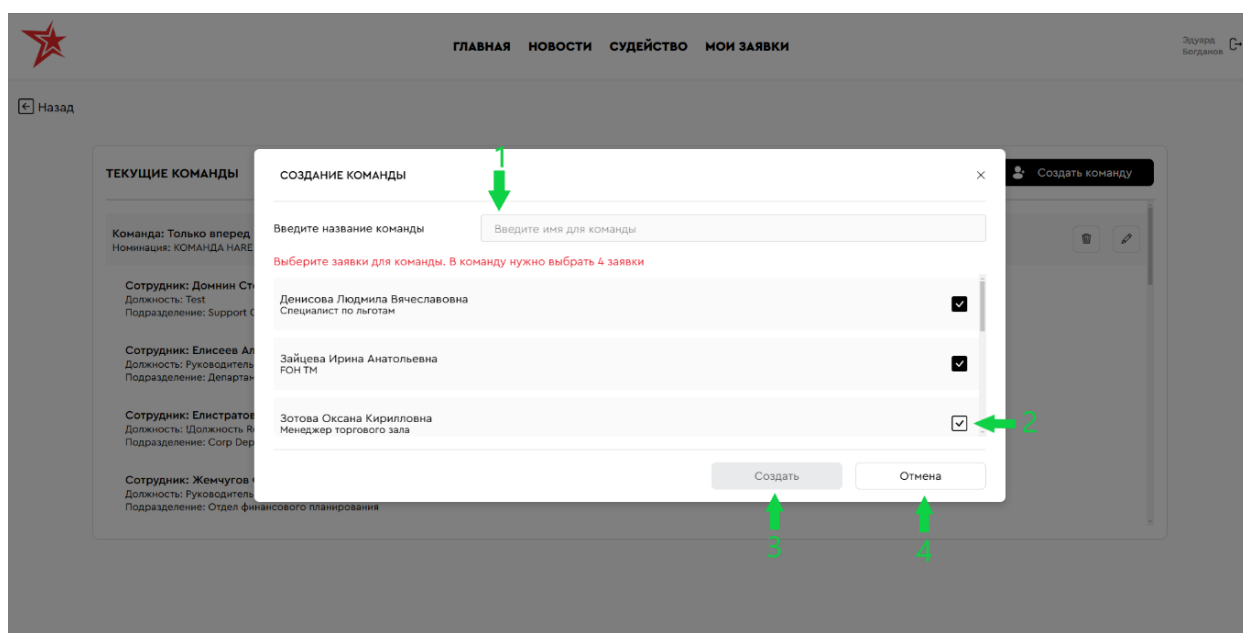


Рисунок 16 – Модальное окно создания команды

Элементы интерфейса модального окна страница управления командой при нажатии на кнопку «Редактировать» представлены на рисунке 17:

- наименование команды (1);
- выбор заявок по командной номинации, включаемых в заявку (2);
- кнопка сохранения команды (3);
- кнопка закрытия модального окна (4).

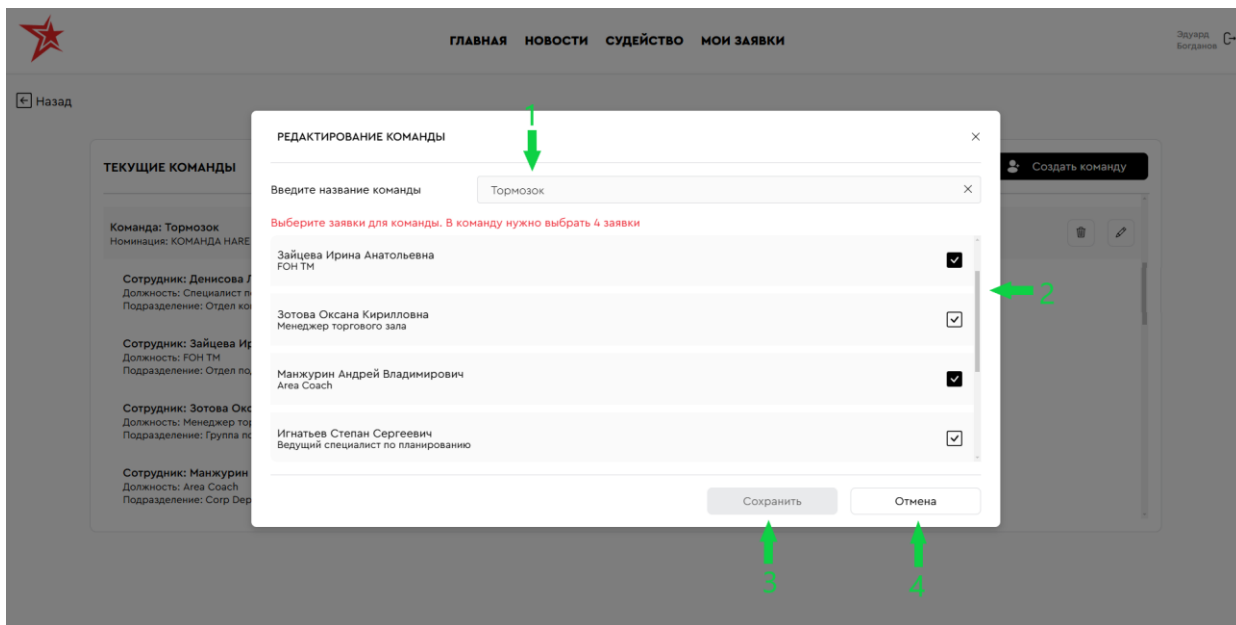


Рисунок 17 – Страница мои заявки

Элемент интерфейса страницы «Судейство» представлены на рисунке 18.

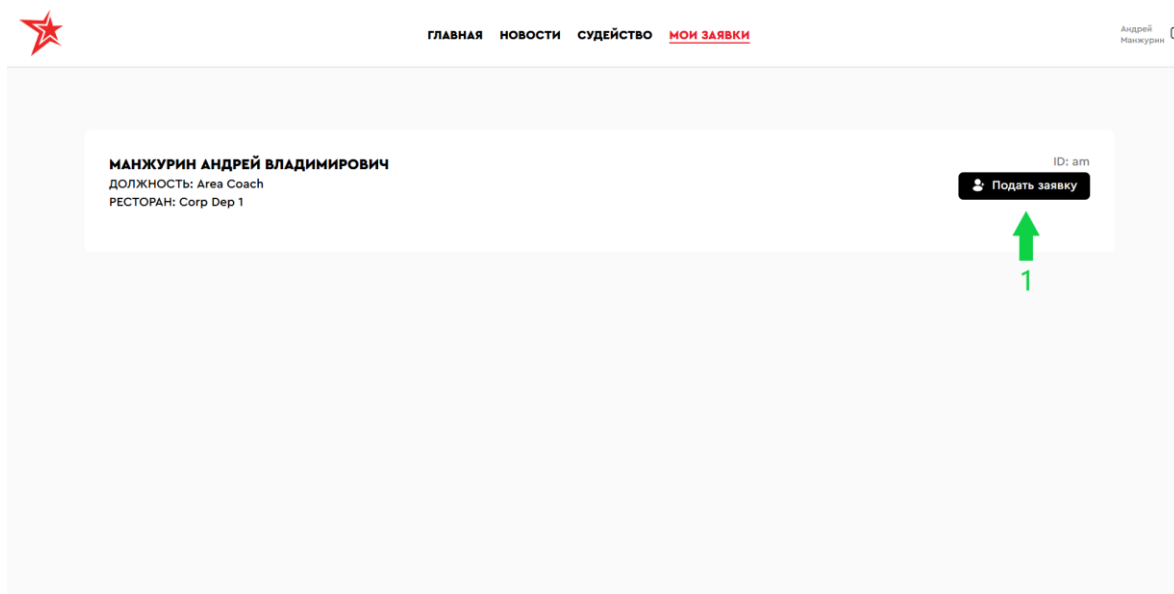


Рисунок 18 – Страница мои заявки

Элементы интерфейса модального окна страницы «Мои заявки» при нажатии на кнопку «Подать заявку» представлены на рисунке 19:

- подтверждение выбора, подача заявки по выбранной номинации (1);
- закрытие модального окна (2).

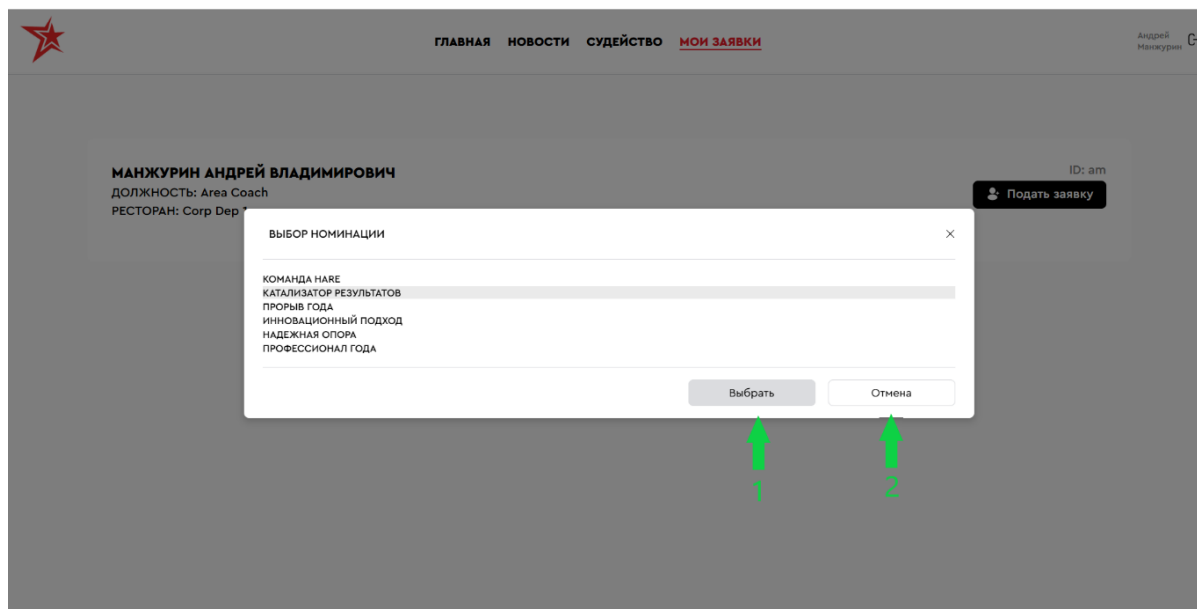


Рисунок 19 – Подача заявки на номинацию

1.4.4 Методика испытаний программы

Функциональное тестирование предполагает проверку всех реализованных функций на предмет соответствия их техническому заданию.

После запуска системы и перехода на вкладку "Оценка" отображается главная страница модуля, информирующая о дате завершения регистрации (рисунок 13).

Для просмотра списка своих заявок следует перейти на вкладку «Мои заявки» (рисунок В.1). Чтобы подать заявку, нужно нажать кнопку «Подать заявку». На экране появится диалоговое окно (рисунок В.2) с выбором номинации. После выбора номинации и нажатия кнопки подтверждения на странице появится уведомление об успешном создании заявки, а также сама созданная заявка со статусом «На рассмотрении» (рисунок В.3). В документе участников появится запись о созданной заявке (рисунок В.4).

При повторной подаче пользователем заявки по той же номинации система должна отказать в создании (рисунок В.5-В.6). Пользователь подал несколько заявок по разным номинациям (рисунок В.7).

Для согласования заявок необходимо авторизоваться под пользователем, который является руководителем подразделения, и перейти на вкладку

«Судейство». На ней появятся заявки, поданные сотрудниками его подразделения, с возможностью одобрить или отклонить каждую (рисунок В.8). Чтобы подтвердить заявку, нужно нажать кнопку «Принять» и в диалоговом окне подтвердить выбор (рисунок В.9). Заявка, поданная по индивидуальной номинации, пропадет из раздела согласования (рисунок В.10) и появится в списке 1 этапа (рисунок В.11). Для смены вкладки номинации следует нажать на название нужной номинации (рисунок В.12).

Чтобы отклонить заявку, нужно нажать кнопку «Отклонить» и в диалоговом окне подтвердить действие. Статус заявки изменится на «Отклонена» (рисунок В.13-В.14). Было подано множество заявок по различным номинациям (рисунок В.15).

При переходе на вкладку командной заявки у пользователя, имеющего доступ к управлению командой, присутствует кнопка «Управление командой» (рисунок В.16). При нажатии на нее пользователь переходит на страницу «Управление командой» (рисунок В.17). Для создания команды нужно нажать кнопку «Создать команду». На странице появится модальное окно с выбором заявок для включения в группу (количество участников – не более 4) и возможностью задать название команде (рисунок В.18). После заполнения полей модального окна следует нажать кнопку создания. На странице появится созданная команда (рисунок В.19-В.20).

Чтобы отредактировать команду, нужно нажать кнопку с изображением карандаша напротив нужной команды. Появится модальное окно, в котором можно изменить состав команды и ее название (рисунок В.21). При подтверждении изменений на странице управления командой обновится название и состав команды (рисунок В.22). Заявки, из которых была сформирована команда, в разделе согласования изменят статус на «В команде» (рисунок В.23).

В разделе первого этапа у пользователя-руководителя подразделения отсутствуют элементы управления заявкой. Это вызвано несовпадением дат проведения. Требуется изменить диапазон дат проведения первого этапа так,

чтобы он включал текущую дату (рисунок В.24-В.25). После этого в разделе первого этапа появится возможность заполнить чек-лист (рисунок В.26). После создания чек-листа пользователь перенаправляется на страницу оценки, где оценивает заявку по компетенциям (рисунок В.27-В.29). По завершении оценки нужно нажать кнопку «Завершить заполнение». Произойдет переадресация на страницу судейства, и в разделе 1 этапа у заявки, для которой был создан чек-лист, появится балл, выставленный руководителем (рисунок В.30). Чек-лист был заполнен повторно, и значение общей оценки скорректировалось (рисунок В.31).

Чтобы перевести заявки на следующий этап, нужно нажать соответствующую кнопку напротив команды и подтвердить действие (рисунок В.32). Чтобы расформировать команду, нужно нажать кнопку «Отклонить» и подтвердить действие. Команда станет серой и отобразит статус "Команда расформирована" (рисунок В.33-В.34).

Команду можно собрать из расформированных заявок. Для этого нужно перейти на страницу управления командой и включить заявки в команду (рисунок В.35-В.37). Если открыть модальное окно создания команды, то в списке заявок будут отсутствовать заявки, уже находящиеся в команде (рисунок В.38). Новая команда была переведена на 2 этап (рисунок В.39).

Чек-листы заполнены территориальным управляющим и руководителем подразделения, но функции управления заявкой доступны только территориальному управляющему (рисунок В.40). Заявка переведена на 3 этап, и у пользователя отсутствует доступ к ней (рисунок В.41-В.42). Расформируем команду, нажав кнопку «Отклонить» (рисунок В.43-В.44).

Сотрудник, являющийся управляющим, имеет возможность заполнить чек-лист (рисунок В.45). Сотруднику приходит уведомление об изменении этапа его заявки (рисунок В.46). Заполнен чек-лист команды (рисунок В.47).

Команду можно формировать из расформированных заявок, которые были расформированы на текущем или предыдущем этапе (рисунок В.48-В.49). Сформируем команду сразу на 3 этап (рисунок В.50). Заполним чек-лист и

переведем команду на финальный этап (рисунок В.51-В.52).

Доступ к оценке сотрудников есть только у пользователей, входящих в группу специалистов по оценке (рисунок В.53). Чек-лист был заполнен специалистами по оценке, возможность управлять командой отсутствует (рисунок В.54-В.55). Заполнен чек-лист, и команда расформирована (рисунок В.56-В.57). Команда переведена на финальный этап, заявка пропала из всех разделов (рисунок В.59).

					ПГУ 1.090501.12.001 ПЗ	Лист
						47
Изм.	Лист	№ докум.	Подпись	Дата		

2 Технико-экономическое обоснование разработки

2.1 Введение

Целью данного раздела является обоснование экономической целесообразности разработки и внедрения модуля многоэтапной оценки компетенций сотрудников в рамках системы Websoft HCM.

Разработка программного обеспечения требует вложений в оплату труда разработчиков, а также определённых накладных расходов. Однако, поскольку модуль предназначен для встраивания в уже существующую ИТ-инфраструктуру, затраты на приобретение дополнительного оборудования не требуются.

Эффективность итогового модуля обусловлено качеством созданного продукта, а также построение эффективного процесса разработки и последующего сопровождения.

В данном разделе дипломного проекта представлены итоговые таблицы, которые позволяют комплексно оценить эффективность разработки по отношению к инвестиционными затратами в нее. Ключевые данные для экономического анализа представлены в таблице 2.1.

Таблица 2.1 – Исходные данные для расчета экономических показателей

Обозначение	Наименование показателя	Единицы измерения	Значение показателя
C_H	Стоимость закупки ЭВМ	Тыс. руб.	60
n_M	Количество рабочих дней в месяце	Дни	22
R_H	Нормативная рентабельность	-	0,25
q_d	Коэффициент доп. заработной платы разработчика	-	0,3
$q_{соц}$	Коэффициент социального страхования	-	0,25
q_n	Коэффициент накладных расходов	-	1,2
w_1	Годовое количество решаемых задач	Зад. год.	45
$t_{м.в}$	Время решения задачи программой	Маш. час.	1
$m_{п}$	Число организаций-покупателей программы	Шт.	2
$T_{эл}$	Стоимость 1 кВт·ч электроэнергии	Руб.	5,4
C_c	Срок эксплуатации модуля	Год	10
НДС	Ставка НДС	%	20

Обозначение	Наименование показателя	Единицы измерения	Значение показателя
K_p	Количество рабочих дней в году	Дни	247
$K_{см}$	Количество рабочих смен ЭВМ	-	2
$t_{см}$	Длительность рабочей смены	Ч	8
A	Процент простоя ЭВМ	%	7
M	Потребляемая мощность ЭВМ	кВт	0,5
$R_{ср}$	Среднее число ремонтов в год	-	1
C_d	Стоимость запчастей для ремонта	Руб.	6000

2.2 Расчет затрат на разработку программы

Общая стоимость разработки модуля складывается из затрат на заработную плату и накладных расходов:

$$C_{рп} = C_{зп} + C_{нак},$$

где $C_{зп}$ – затраты по заработной плате инженера-программиста;

$C_{нак}$ – накладные расходы.

Заработная плата программиста рассчитывается по следующей формуле:

$$C_{зп} = ОЗП * (1 + C_{соц}) * (1 + q_d) * t_m,$$

где ОЗП – основная заработная плата инженера-программиста за месяц (30000 руб.);

t_m – трудоемкость разработки в человеко-месяцах;

q_d – коэффициент дополнительной заработной платы;

$q_{соц}$ – коэффициент отчислений на соц. страхование.

Время разработки составило 45 дней. При $n_m = 22$ рабочих дня в месяце:

$$t_m = 45/22 = 2,05 \text{ (чел. –мес.)}.$$

Таким образом, затраты по заработной плате инженера-программиста составят:

$$C_{зп} = 30\,000 * (1 + 0,25) * (1 + 0,3) * 2,05 = 99\,843,75 \text{ (руб.)}.$$

Накладные затраты рассчитываются с учетом q_n [1] – коэффициента, определяющего уровень накладных расходов организации по следующей формуле:

$$C_{нак} = ОЗП * q_n * t_m,$$

$$C_{нак} = 30\,000 * 1,2 * 2,05 = 73\,800 \text{ (руб.)}.$$

Таким образом, суммарные затраты на разработку программы составляют:

$$C_{\text{рп}} = 99\,843,75 + 73\,800 = 173\,643,75 \text{ (руб.)}.$$

2.3 Расчет цены разработанной программы

Оптовая цена разработанной программы определяется [7] по следующей формуле:

$$\text{ОЦ} = C_{\text{рп}} + (1 + R_{\text{н}}),$$

где ОЦ – оптовая цена (цена разработчика) (руб.);

$C_{\text{рп}}$ – суммарные затраты на разработку программы (руб.);

$R_{\text{н}}$ – норматив рентабельности, учитывающий прибыль организации, разрабатывающей данную программу в долях ко всем затратам данной организации на разработку программы.

$$\text{ОЦ} = 173\,643,75 * (1 + 0,25) = 217\,054,69 \text{ (руб.)}.$$

Розничная цена программы рассчитывается с учетом налога на добавленную стоимость (НДС = 20%) по следующей формуле:

$$\text{РЦ} = \text{ОЦ} + \text{НДС} = \text{ОЦ} + \text{ОЦ} * 0.2 = \text{ОЦ} * 1.2,$$

$$\text{РЦ} = 217\,054,69 * 1,2 = 260\,465,63 \text{ (руб.)}.$$

Выручка от продаж при условии $m_{\text{п}}$ - количество организаций, желающих приобрести программу, составит:

$$B_{\text{пр}} = \text{РЦ} * m_{\text{п}},$$

$$B_{\text{пр}} = 260\,465,63 * 2 = 520\,931,26 \text{ (руб.)}$$

2.4 Расчет капитальных вложений

Капиталовложения, связанные с работой ЭВМ рассчитываются по следующей формуле:

$$C_{\text{ЭВМ}} = C_{\text{н}} + B_{\text{по}},$$

где $C_{\text{н}}$ – стоимость ЭВМ (руб.);

$C_{\text{по}}$ – стоимость программного обеспечения, необходимого для разработки устройства.

					ПГУ 1.090501.12.001 ПЗ	Лист
						50
Изм.	Лист	№ докум.	Подпись	Дата		

Примерная конфигурация ПК:

- процессор: Intel Core i7-3,2 Гц;
- ОЗУ: 16гб;
- видеокарта: Nvidia GeForce GTX2060-6144 Мб;
- носитель: SSD 512 Gb.

Программа разрабатывается на языке JavaScript, на языке разметки HTML, на языке таблиц стилей CSS и с использованием библиотеки React.

Поэтому $C_{по} = 20\,000$.

Капиталовложения в ЭВМ составляют:

$$C_{ЭВМ} = 60\,000 + 20\,000 = 80\,000 \text{ (руб.)}.$$

2.5 Расчет эксплуатационных расходов

Расходы на эксплуатацию программы и системы, в комплексе с которой она работает, несет покупатель.

Эксплуатационные расходы на ЭВМ рассчитываются по следующей формуле:

$$P_э = \frac{(T_{МВ} * P_ч) + P_ц}{C_c},$$

где $T_{МВ}$ – машинное время для решения задач с помощью разработанной программы, (маш.час/год);

$P_ч$ – эксплуатационные расходы, приходящиеся на 1 час работы ЭВМ;

$P_ц$ – цена, по которой продается программа (руб.);

C_c – срок службы программы (г).

Полезный фонд времени работы ЭВМ рассчитывается по следующей формуле:

$$\Phi_{пол} = \Phi_{общ} * t_{см} * (1 - a / 100),$$

где $\Phi_{общ}$ – общий фонд времени работы ЭВМ (дни); $\Phi_{общ} = K_p$;

$t_{см}$ – время одного рабочего дня (час);

a – простои ЭВМ (в % от общего фонда времени работы ЭВМ).

Полезный фонд времени работы ЭВМ получим:

					ПГУ 1.090501.12.001 ПЗ	Лист
						51
Изм.	Лист	№ докум.	Подпись	Дата		

$$\Phi_{\text{пол}} = 247 * 8 * (1 - 7 / 100) = 1837,68.$$

Машинное время для решения задач с помощью данной программы рассчитывается по следующей формуле:

$$T_{\text{м.в}} = w_I * t_{\text{мв}},$$

где w_I – количество задач, решаемых потребителем в год(шт.);

$T_{\text{м.в}}$ – время решения задачи, разработанной программой (маш.час).

$$T_{\text{м.в}} = 45 * 1 = 45 \text{ (маш.час / год)}.$$

Эксплуатационные расходы, приходящиеся на 1 час работы ЭВМ, оцениваются по следующей формуле:

$$P_{\text{ч}} = \frac{(P_a + P_{\text{зп}} + P_{\text{эл}} + P_{\text{рм}})}{\Phi_{\text{пол}}},$$

где P_a – амортизационные отчисления (руб.);

$P_{\text{зп}}$ – затраты по заработной плате инженера в год (руб./год);

$P_{\text{эл}}$ – стоимость потребляемой энергии (руб.);

$P_{\text{рм}}$ – затраты на ремонт ЭВМ (руб.);

$\Phi_{\text{пол}}$ – полезный годовой фонд работы ЭВМ, (маш.час/год).

Амортизационные отчисления рассчитываются с учетом нормы амортизации ($a_H = 10 \%$);

$$P_a = a_H * C_{\text{ЭВМ}} = 0,1 * 80\,000 = 8\,000 \text{ (руб.)}.$$

Затраты по заработной плате инженера за год рассчитывается по следующей формуле:

$$P_{\text{зп}} = (1 + q_{\text{соц}}) * (1 + q_d) * \text{ОЗП} * 12,$$

где $q_{\text{соц}}$ – коэффициент, учитывающий начисления органам социального страхования на заработную плату разработчика программы, в долях к сумме основной заработной плате разработчика;

q_d – коэффициент, учитывающий дополнительную заработную плату разработчика программы, в долях к сумме основной заработной платы;

ОЗП – основная заработная плата инженера-программиста за месяц.

Рассчитываем годовые затраты по заработной плате и социальным отчислениям для инженера-программиста:

$$P_{3П} = (1+0,25) * (1+0,3) * 30\,000 * 12 = 585\,000 \text{ (руб./ год.)}.$$

Стоимость потребляемой энергии оценивается по следующей формуле:

$$P_{ЭЛ} = M * \Phi_{Пол} * T_{ЭЛ},$$

где M – мощность, потребляемая ЭВМ (кВт);

$\Phi_{Пол}$ – полезный годовой фонд работы ЭВМ (маш. час/год);

$T_{ЭЛ}$ – тариф за 1 кВт/час (руб. /кВт).

Произведем расчет стоимости потребляемой энергии:

$$P_{ЭЛ} = 0,5 * 1837,68 * 5,4 = 4\,961,74 \text{ (руб.)}.$$

Затраты на ремонт ЭВМ вычисляются по следующей формуле:

$$P_{РМ} = R_{СР} * C_{Д} ,$$

где $R_{СР}$ – среднее количество ремонтов в год;

$C_{Д}$ – стоимость деталей, заменяемых при одном ремонте, в среднем.

$$P_{РМ} = 1 * 6\,000 = 6\,000 \text{ (руб.)}.$$

Произведем вычисление эксплуатационных расходов, приходящихся на 1 час работы ЭВМ:

$$P_{ч} = \frac{(8\,000 + 585\,000 + 4\,961,74 + 6000)}{1\,837,68} = 323,79 \text{ (руб./час)}$$

Далее вычислим эксплуатационные расходы на ЭВМ:

$$P_{Э} = \frac{(45 * 323,79) + 259\,335}{10} = 27\,390,56 \text{ (руб.)}.$$

2.6 Расчет денежного годового экономического эффекта

Денежный годовой экономический эффект оценивается по следующей формуле:

$$\Gamma_{Э} = M_{Э} + k_{Н} * K_{Э},$$

где $M_{Э}$ – экономия стоимости машинного времени, руб.;

$k_{Н}$ – нормативный коэффициент эффективности капитальных вложений;

$k_{Н} = 0,2$;

$K_{Э}$ – экономия капитальных вложений, руб..

Экономия капитальных вложений рассчитывается по следующей формуле:

					ПГУ 1.090501.12.001 ПЗ	Лист
						53
Изм.	Лист	№ докум.	Подпись	Дата		

$$K_{\text{Э}} = \frac{T_{\text{М.В}} * C_{\text{ЭВМ}}}{\Phi_{\text{ПОЛ}}},$$

где $T_{\text{М.В}}$ – машинное время для решения задач с помощью разработанной программы (маш. час/год);

$C_{\text{ЭВМ}}$ – капиталовложения в ЭВМ (руб.);

$\Phi_{\text{ПОЛ}}$ – полезный годовой фонд работы ЭВМ (маш. час/год).

$$T_{\text{М.В}} = 45 \text{ маш. час/год.}$$

Произведем расчет экономии капитальных вложений по следующей формуле:

$$K_{\text{Э}} = \frac{45 * 80\,000}{1\,837,68} = 1\,959,68 \text{ (руб.)},$$

Расчет экономии стоимости машинного времени производится по следующей формуле:

$$M_{\text{Э}} = P_{\text{ч}} * T_{\text{М.В}},$$

где $P_{\text{ч}}$ – эксплуатационные расходы, приходящиеся на 1 час работы ЭВМ;

$T_{\text{М.В}}$ – машинное время для решения задач с помощью разработанной программы (маш. час/год);

$$M_{\text{Э}} = 323,79 * 45 = 14\,570,55 \text{ (руб./год)},$$

$$\Gamma_{\text{Э}} = 14\,570,55 + 0,2 * 1\,959,68 = 14\,962,49 \text{ (руб./год)}.$$

2.7 Итоговые расходы

Обобщенные технико-экономические показатели разработки программы сведены в таблицу 2.7.2.

Программа разрабатывалась в течение 45 календарных рабочих дней. Учитывая, что в среднем в месяце 22 рабочий день, трудоемкость составила примерно 2,05 человеко-месяца, что соответствует оплате труда за 2 месяца и 1 неделю. Таким образом, затраты на заработную плату программисту составили 99 450 рублей.

Дополнительно к этому были учтены капитальные вложения в оборудование, эксплуатационные расходы и затраты на разработку программного модуля. Все значения приведены в таблице ниже.

					ПГУ 1.090501.12.001 ПЗ	Лист
						54
Изм.	Лист	№ докум.	Подпись	Дата		

Таблица 2.7.2 – Обобщённые технико-экономические показатели разработки

Наименование показателя	Значение показателя (руб.)
Капиталовложение в ЭВМ	80 000
Затраты на разработку модуля	172 890
Эксплуатационные затраты	27 390,56
Затраты на заработную плату программисту (руб.)	99 450
Итоговая стоимость	379 730,56

3 Охрана труда

Длительная работа за персональным компьютером (ПЭВМ) может негативно сказываться на здоровье пользователя. Даже кратковременная работа на компьютере, установленном с грубыми нарушениями норм, приводит к утомлению. Неправильно организованное рабочее место оказывает вредное воздействие на зрение, опорно-двигательный аппарат и общее психофизиологическое состояние организма. Поэтому строгое соблюдение мер безопасности жизнедеятельности является обязательным для предотвращения негативных последствий.

При выборе помещения для работы с ПЭВМ предпочтение следует отдавать комнатам, ориентированным на север или северо-восток. Категорически не рекомендуется размещать рабочие места в цокольных или подвальных этажах. Минимальная площадь на одно рабочее место составляет 6 м² для компьютеров с ЭЛТ-мониторами и 4,5 м² – с ЖК (плазменными) мониторами. Для защиты от прямых солнечных лучей окна должны быть оборудованы солнцезащитными жалюзи или однотонными шторами из плотной звукопоглощающей ткани. Цвет штор должен гармонировать со стенами, а их ширина – вдвое превышать ширину окна для формирования складок. В помещении обязательна ежедневная влажная уборка (включая удаление пыли с мониторов, столов, подоконников) и систематическое проветривание после каждого часа работы.

Мебель для обивки должна изготавливаться из полимерных материалов, имеющих санитарно-эпидемиологические заключения, чтобы предотвратить выделение в воздух вредных веществ от разложения синтетических и лакокрасочных материалов. Не рекомендуется использовать древесностружечные плиты (ДСП), ламинированные бумагопласты и синтетические ковровые покрытия. Рабочий стол должен обеспечивать пространство для ног: высота не менее 600 мм, ширина не менее 500 мм, глубина на уровне колен не менее 450 мм, на уровне вытянутых ног не менее

					ПГУ 1.090501.12.001 ПЗ	Лист
						56
Изм.	Лист	№ докум.	Подпись	Дата		

650 мм. Для комфортной работы необходима подставка для ног шириной не менее 300 мм, глубиной не менее 400 мм, с регулировкой высоты до 150 мм и наклоном опорной поверхности до 20°. Поверхность подставки должна быть текстурированной и иметь передний бортик высотой 10 мм.

Рабочее место пользователя ПЭВМ должно быть оснащено необходимым оборудованием (стол, стул, монитор, клавиатура, мышь), соответствующим стандартам безопасности и эргономики. Расстояние между столами с мониторами – не менее 2,0 м, между боковыми поверхностями мониторов – не менее 1,2 м. Рабочая поверхность стола должна иметь высоту 680-800 мм (оптимально 725 мм при отсутствии регулировки). Конструктивные размеры столешницы рекомендуются из стандартных модулей: ширина 800, 1000, 1200 или 1400 мм, глубина 800 или 1000 мм. Монитор следует располагать так, чтобы естественный свет падал преимущественно слева; допустимо размещение в плоскости окна. Экран должен находиться на расстоянии 600-700 мм от глаз (не ближе 500 мм). Клавиатура размещается на расстоянии 100-300 мм от края стола, обращенного к пользователю, или на отдельной регулируемой по высоте поверхности. Рабочее место должно иметь удобный доступ к розеткам и оборудованию для безопасной прокладки кабелей, а также средства защиты от излучений и перегрева (экранирующие устройства, системы вентиляции).

Офисный стул должен поддерживать оптимальную рабочую позу, позволять менять положение тела для снижения напряжения мышц шеи и плеч, давать отдых спине и предотвращать утомление. Выбор стула учитывает рост пользователя, специфику работы и продолжительность работы за ПК. Стул обязан иметь функции подъема, поворота, независимую регулировку высоты сиденья (400-550 мм), угла наклона сиденья (вперед до 15°, назад до 5°), высоты и угла наклона спинки ($\pm 30^\circ$ от вертикали), а также расстояния от спинки до переднего края сиденья (260-400 мм). Все регулировки должны быть удобными и надежно фиксироваться. Сиденье шириной и глубиной не менее 400 мм обязано иметь закругленный передний край. Спинка должна иметь

опорную поверхность высотой 300 ± 20 мм, шириной не менее 380 мм и радиус кривизны в горизонтальной плоскости около 400 мм. Подлокотники (стационарные или съемные) длиной не менее 250 мм и шириной 50-70 мм должны регулироваться по высоте над сиденьем (230 ± 30 мм) с внутренним расстоянием между ними 350-500 мм. Материалы сиденья и спинки – мягкие, с низким трением, слабой электростатичностью, воздухопроницаемые и легкоочищаемые.

Персонал, ответственный за организацию рабочих мест, обязан регулярно проводить проверки и обслуживание оборудования для обеспечения его безопасности и работоспособности. Пользователи ПЭВМ должны проходить обучение правилам безопасной работы, включая правильную посадку, необходимость регулярных перерывов, выполнение упражнений для глаз и спины, а также правила обработки информации и обеспечения конфиденциальности. Для задач, требующих постоянного взаимодействия с ПК и высокой концентрации, рекомендуется делать перерывы продолжительностью 10-15 минут каждые 45-60 минут работы. Сотрудникам, интенсивно использующим компьютеры, необходим отдых и снятие стресса в специализированных помещениях для психологической разгрузки в течение дня и по окончании работы. Предлагается также провести обучение базовым навыкам работы с ПЭВМ, регулярно обновлять оборудование, создать централизованное хранилище данных, установить ПО для мониторинга и безопасности, проводить регулярные аудиты информационной безопасности.

При работе с ПЭВМ на здоровье работников могут воздействовать следующие негативные факторы:

Плохая эргономика: Неправильное расположение монитора, клавиатуры, мыши, отсутствие регулировок стола/стула ведет к болям в спине, шее, запястьях.

Излучение монитора: Длительное воздействие вызывает усталость глаз, головные боли, ухудшение зрения.

Монотонность: Постоянное повторение движений и задач приводит к

					ПГУ 1.090501.12.001 ПЗ	Лист
						58
Изм.	Лист	№ докум.	Подпись	Дата		

утомлению и стрессу.

Сидячий образ жизни: Способствует ожирению, сердечно-сосудистым заболеваниям, проблемам с суставами и мышцами.

Психологические аспекты: Высокие требования, отсутствие отдыха, конфликт ценностей вызывают стресс и депрессию.

Информационная перегрузка: Постоянный поток информации затрудняет концентрацию и вызывает утомление.

Недостаток движения: Работа за ПК ограничивает физическую активность, повышая риск различных заболеваний.

					ПГУ 1.090501.12.001 ПЗ	Лист
						59
Изм.	Лист	№ докум.	Подпись	Дата		

Заключение

В данной работе был разработан программный модуль, предназначенный для автоматизации многоэтапной оценки компетенций сотрудников в системе управления персоналом Websoft HCM. В ходе исследования были проанализированы современные подходы и системы в области подбора и оценки персонала, выявлены их преимущества и недостатки, что позволило обосновать актуальность создания нового модуля.

Разработанное решение включает серверное приложение, реализующее бизнес-логику на платформе Websoft HCM, и клиентскую часть на React.js, обеспечивающую интуитивно понятный пользовательский интерфейс. Модуль поддерживает ключевые функции: управление заявками, формирование команд, проведение оценки с использованием чек-листов, а также ролевое разграничение доступа и систему уведомлений. Все данные надежно сохраняются в реляционной СУБД MS SQL Server, интегрированной с системой документооборота и корпоративным порталом.

В рамках реализации были определены функциональные и технические требования, спроектирована архитектура модуля с использованием RPC-паттерна и реализована база данных. Проведены тестирования, подтвердившие корректность работы всех компонентов системы. В качестве среды разработки использовался редактор Visual Studio Code, а взаимодействие между клиентом и сервером организовано с применением формата JSON.

Разработанное приложение имеет высокую практическую ценность, так как позволяет существенно повысить эффективность кадровых процессов, обеспечить прозрачность оценки, снизить влияние человеческого фактора и стандартизировать процесс подбора персонала. Модуль может быть внедрён в крупные и средние организации и обладает потенциалом для масштабирования и дальнейшего развития.

					ПГУ 1.090501.12.001 ПЗ	Лист
						60
Изм.	Лист	№ докум.	Подпись	Дата		

Список использованных источников

1. Савицкая Г.В. Анализ хозяйственной деятельности предприятия. – Минск: Новое знание, 2020. – 688 с.
2. Таненбаум Э.С., Ван Стеен М. Распределенные системы. Принципы и парадигмы. – СПб.: Питер, 2016. – 608 с.
3. Учебное пособие по Python [Электронный ресурс]. – Режим доступа: <https://docs.python.org/3/tutorial/index.html> (дата обращения: 06.06.2025).
4. PHP Manual – Официальная документация [Электронный ресурс]. – Режим доступа: <https://www.php.net/manual/ru/introduction.php> (дата обращения: 06.06.2025).
5. Harver. Official Website [Электронный ресурс]. – Режим доступа: <https://www.harver.com/> (дата обращения: 01.06.2025).
6. Websoft HCM – Официальный сайт [Электронный ресурс]. – Режим доступа: <https://websoft.ru/> (дата обращения: 01.06.2025).
7. SAP SuccessFactors – Обзор [Электронный ресурс]. – Режим доступа: <https://www.sap.com/products/hcm.html> (дата обращения: 02.06.2025).
8. Talantix – Платформа для подбора персонала [Электронный ресурс]. – Режим доступа: <https://talantix.ru/> (дата обращения: 03.06.2025).
9. Когда использовать базу данных MS SQL Server [Электронный ресурс]. – Режим доступа: <https://www.dbserve.ru/blog/kogda-ispolzovat-bazu-dannyh-ms-sql-server> (дата обращения: 02.06.2025).
10. Mozilla Developer Network. JavaScript Guide [Электронный ресурс]. – Режим доступа: <https://developer.mozilla.org/> (дата обращения: 03.06.2025).

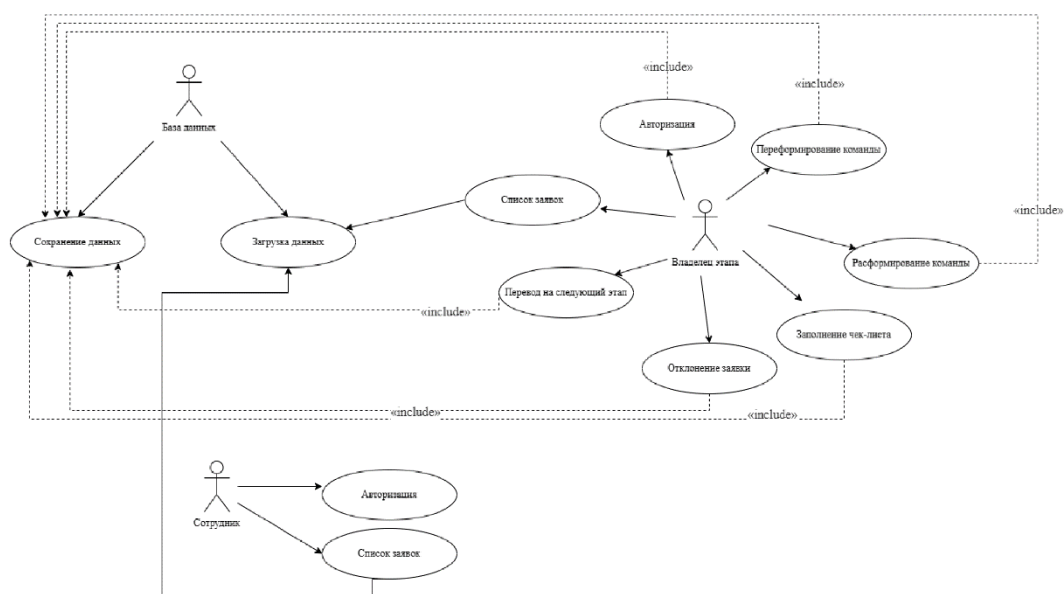
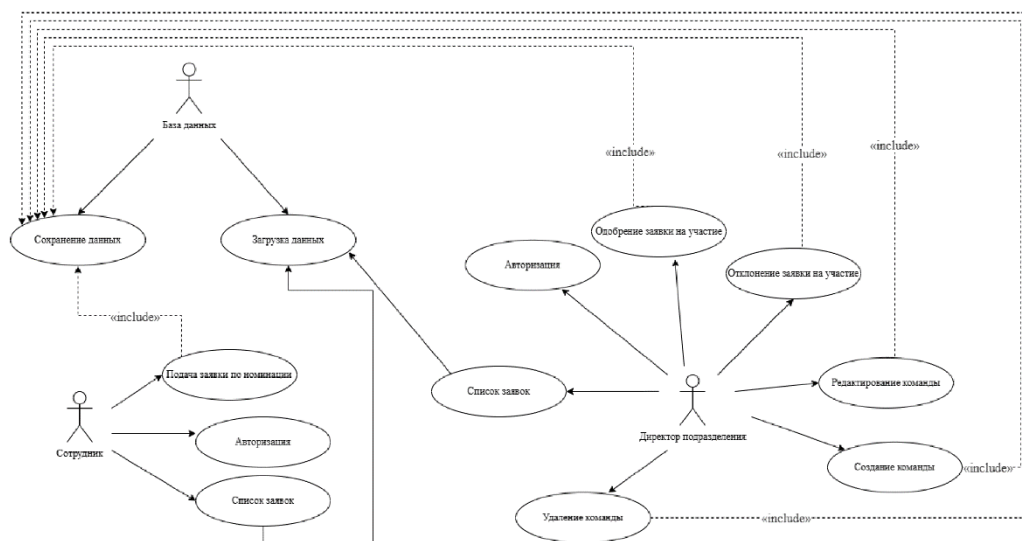
Приложение А

(обязательное)

Модуль проведения многоэтапного отбора автоматизированной системы
управления персоналом
UML-диаграммы

					ПГУ 1.090501.12.001 ПЗ	Лист
						62
Изм.	Лист	№ докум.	Подпись	Дата		

Приложение А.1 – Диаграмма вариантов использования



№ 1.090501.12.001.019									
Изм.	Лист	№ докум.	Подпись	Дата	Модуль: производство технологического оборудования				
Разработчик	Проверен	Модифицирован	Утвержден	Дата	Модуль: автоматизированная система управления производством				
Лист	Листов	Листов	Листов	Листов	ИИ: директор, директор по инновациям				
Лист	Листов	Листов	Листов	Листов	Лист				
Лист	Листов	Листов	Листов	Листов	Лист				
Лист	Листов	Листов	Листов	Листов	Лист				
Лист	Листов	Листов	Листов	Листов	Лист				
Лист	Листов	Листов	Листов	Листов	Лист				

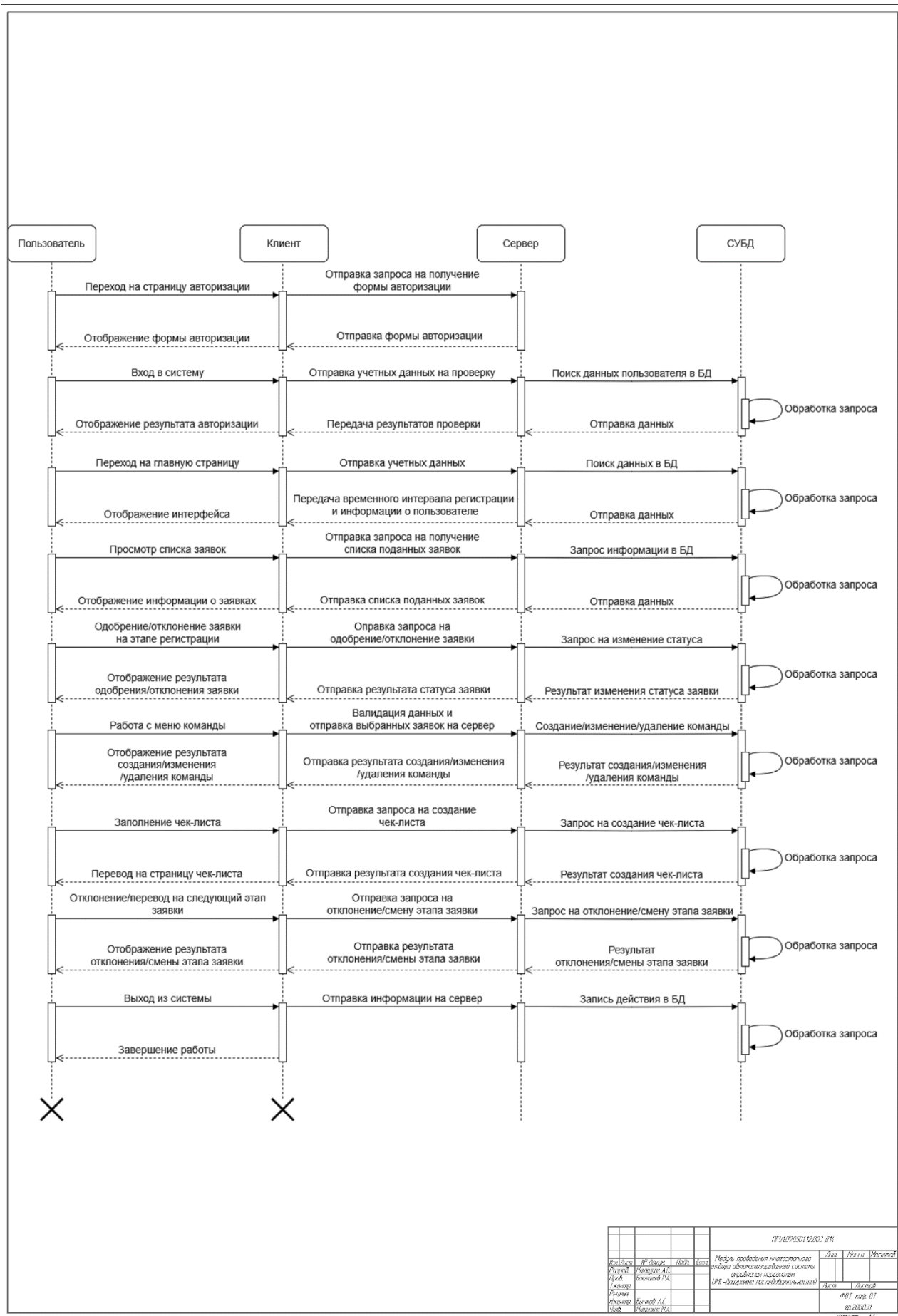
Изм.	Лист	№ докум.	Подпись	Дата

ПГУ 1.090501.12.001 ПЗ

Лист

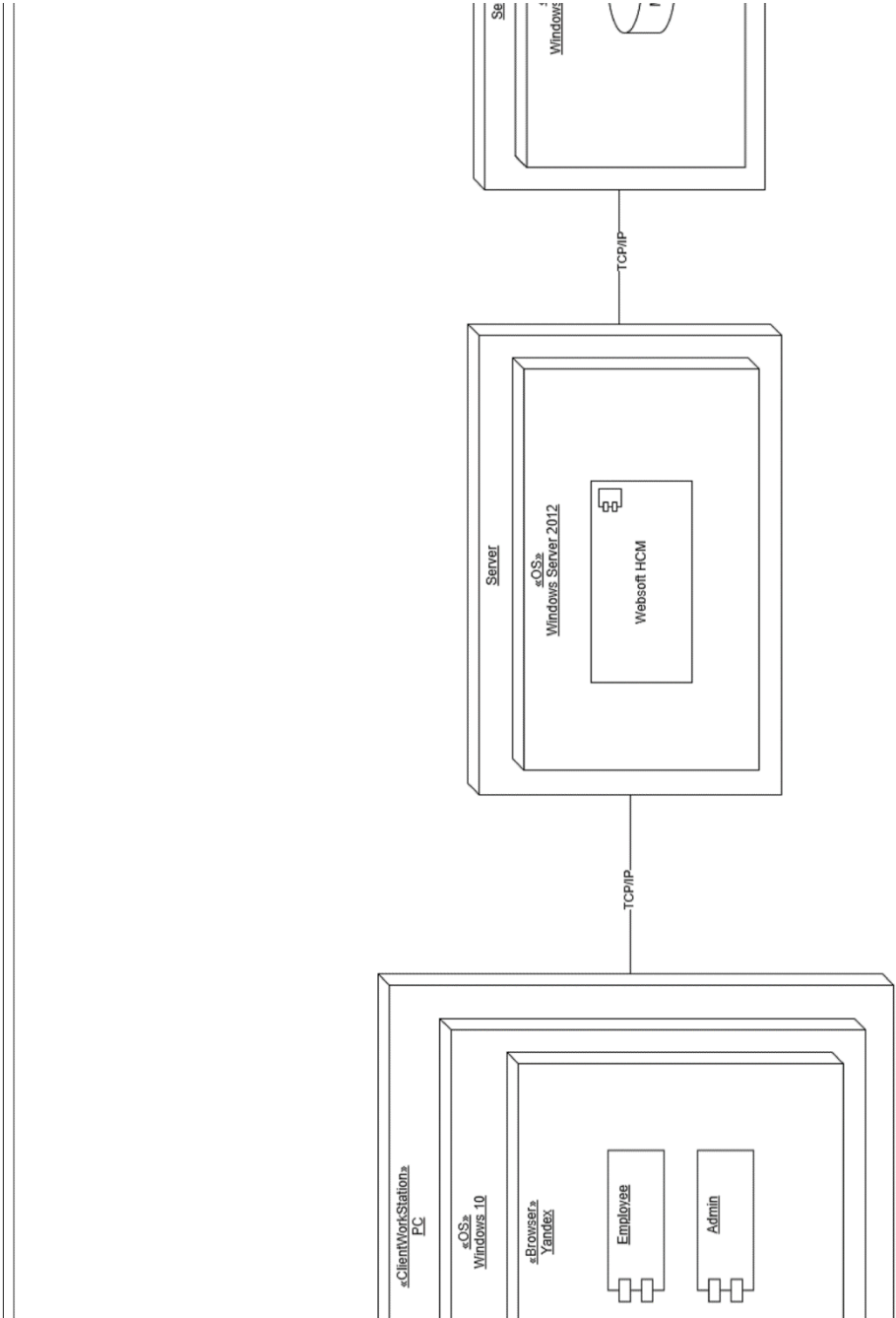
63

Приложение А.2 – Диаграмма последовательностей

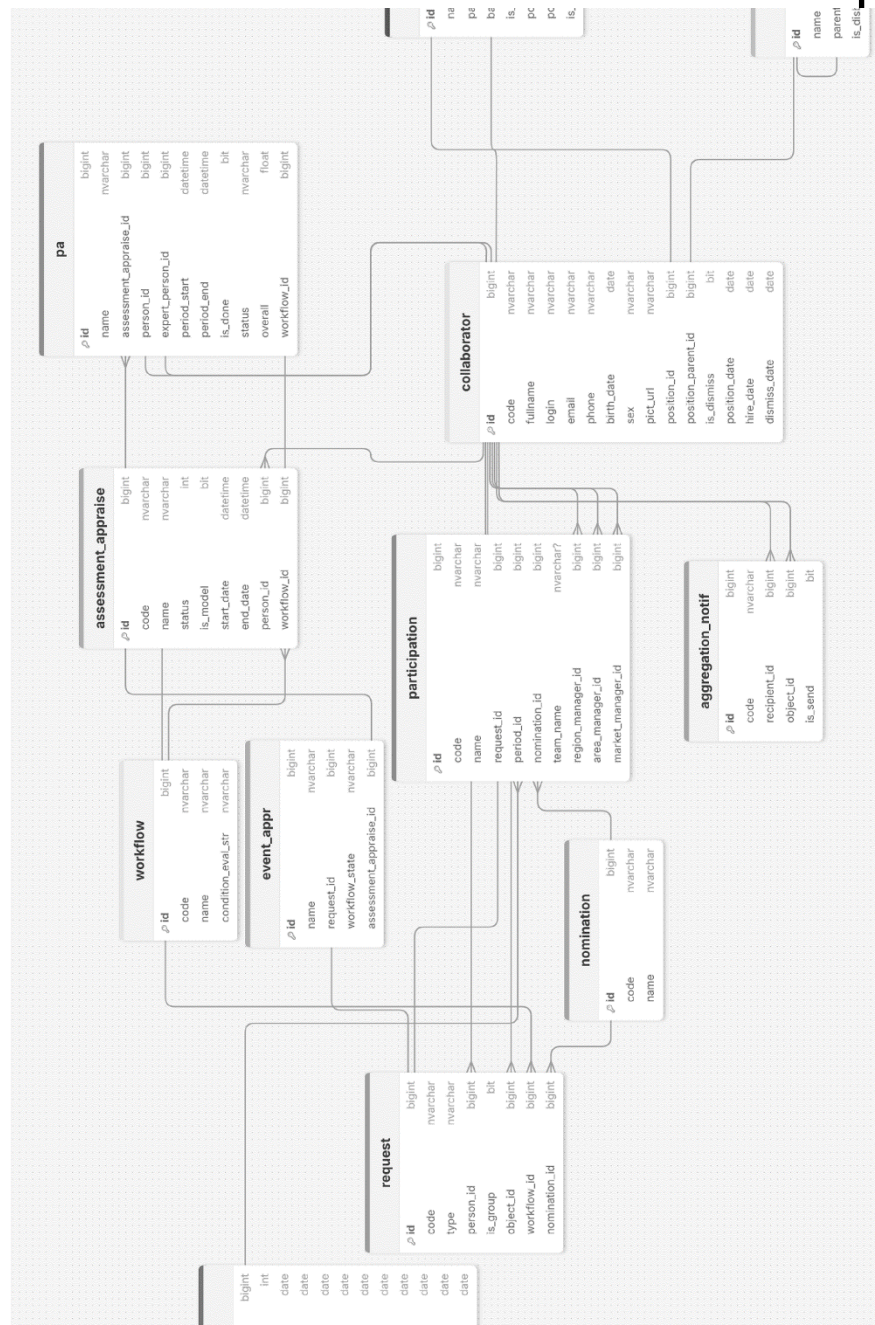


ПГУ 1.090501.12.001 ПЗ									
Исполн.	И.И. Иванов	Провер.	И.И. Иванов	Модуль, разработчик	И.И. Иванов	Модуль, разработчик	И.И. Иванов	Модуль, разработчик	И.И. Иванов
Провер.	И.И. Иванов	Провер.	И.И. Иванов	Модуль, разработчик	И.И. Иванов	Модуль, разработчик	И.И. Иванов	Модуль, разработчик	И.И. Иванов
Провер.	И.И. Иванов	Провер.	И.И. Иванов	Модуль, разработчик	И.И. Иванов	Модуль, разработчик	И.И. Иванов	Модуль, разработчик	И.И. Иванов
Провер.	И.И. Иванов	Провер.	И.И. Иванов	Модуль, разработчик	И.И. Иванов	Модуль, разработчик	И.И. Иванов	Модуль, разработчик	И.И. Иванов
Провер.	И.И. Иванов	Провер.	И.И. Иванов	Модуль, разработчик	И.И. Иванов	Модуль, разработчик	И.И. Иванов	Модуль, разработчик	И.И. Иванов
Провер.	И.И. Иванов	Провер.	И.И. Иванов	Модуль, разработчик	И.И. Иванов	Модуль, разработчик	И.И. Иванов	Модуль, разработчик	И.И. Иванов
Провер.	И.И. Иванов	Провер.	И.И. Иванов	Модуль, разработчик	И.И. Иванов	Модуль, разработчик	И.И. Иванов	Модуль, разработчик	И.И. Иванов
Провер.	И.И. Иванов	Провер.	И.И. Иванов	Модуль, разработчик	И.И. Иванов	Модуль, разработчик	И.И. Иванов	Модуль, разработчик	И.И. Иванов
Провер.	И.И. Иванов	Провер.	И.И. Иванов	Модуль, разработчик	И.И. Иванов	Модуль, разработчик	И.И. Иванов	Модуль, разработчик	И.И. Иванов

Приложение А.3 – Диаграмма развертывания



Приложение А.4 – Физическая модель данных



Приложение Б

(обязательное)

Модуль проведения многоэтапного отбора автоматизированной системы
управления персоналом
Листинги программы

Файл custom_skill_tools.js

```
function getCurrentPeriodId(year) {
    year = OptDate(year);

    if (year == undefined) {
        year = Year(Date());
    }

    sql = "SELECT id FROM cc_skill_periods WHERE year = " +
year;

    return ArrayOptFirstElem(XQuery(sql), { id: undefined
}).id;
}

function getCurrentPeriod(year) {
    year = OptDate(year);

    if (year == undefined) {
        year = Year(Date());
    }

    sqlReq = " DECLARE @year int = " + year + "; " +
        " SELECT " +
        "     x.id, " +
        "     x.title, " +
        "     x.workflow_stage_code, " +
        "     x.title_full, " +
        "     x.description, " +
        "     x.start_date, " +
        "     x.end_date, " +
        "     x.img_id " +
        " FROM cc_skill_periods p " +
        " CROSS APPLY ( " +
        "     VALUES " +
        "         (1, 'Регистрация',      'approving',
p.request_title, p.request_description, p.request_start_date,
p.request_finish_date, p.request_img), " +
        "         (2, 'Этап 1',          'stage_1',
p.stage_1_title, p.stage_1_description, p.stage_1_start_date,
p.stage_1_finish_date, p.stage_1_img), " +
        "         (3, 'Этап 2',          'stage_2',
p.stage_2_title, p.stage_2_description, p.stage_2_start_date,
p.stage_2_finish_date, p.stage_2_img), " +
        "         (4, 'Этап 3',          'stage_3',
p.stage_3_title, p.stage_3_description, p.stage_3_start_date,
p.stage_3_finish_date, p.stage_3_img), " +
        "         (5, 'Этап 4',          'stage_4',
p.stage_4_title, p.stage_4_description, p.stage_4_start_date,
p.stage_4_finish_date, p.stage_4_img) " +
        " ) AS x(id, title, workflow_stage_code, title_full,
description, start_date, end_date, img_id) " +
```

```

        " WHERE p.year = @year " +
        " ORDER BY p.id, x.start_date; ";

    var result = ArraySelectAll(XQuery("sql: " + sqlReq));
    if (ArrayOptSize(result) == 0) throw "Не найдено лиги за "
+ year + " год"

    return result;
}

function getBosses(coll_id) {
    var sql_req = "sql: " +
        " DECLARE @coll_id BIGINT = " + OptInt(coll_id, 0) +
"; " +
        " WITH tree AS ( " +
        "     SELECT " +
        "         fms.object_id 'coll_id', " +
        "         fms.person_id 'boss_id', " +
        "         fms.person_fullname 'boss_fullname', " +
        "         fms_cls.position_name
'boss_position_name', " +
        "         0 AS level " +
        "     FROM func_managers fms " +
        "     INNER JOIN collaborators fms_cls ON fms_cls.id
= fms.person_id " +
        "     WHERE fms.object_id = @coll_id " +
        "     UNION ALL " +
        "     SELECT " +
        "         fms.object_id 'coll_id', " +
        "         fms.person_id 'boss_id', " +
        "         fms.person_fullname 'boss_fullname', " +
        "         fms_cls.position_name
'boss_position_name', " +
        "         t.level + 1 AS level " +
        "     FROM tree t " +
        "     INNER JOIN func_managers fms ON fms.object_id =
t.boss_id " +
        "     INNER JOIN collaborators fms_cls ON fms_cls.id
= fms.person_id " +
        "     WHERE t.boss_id IS NOT NULL " +
        " ) " +
        " SELECT " +
        "     boss_id 'id', " +
        "     boss_fullname 'fullname', " +
        "     boss_position_name, " +
        "     CASE " +
        "         WHEN level = 0 THEN 'RGM' " +
        "         WHEN level = 1 THEN 'Area Coach' " +
        "         WHEN level = 2 THEN 'Market Coach' " +
        "         WHEN boss_position_name = 'RSC' THEN 'RSC' " +
        "     END 'level' " +
        " FROM tree "

```

```

        return ArraySelectAll(XQuery(sql_req));
    }

    function isBossForCollaborator(coll_id, boss_id) {
        var bosses = getBosses(coll_id);
        if (ArrayOptFind(bosses, 'This.id == ' + OptInt(boss_id,
0)) == undefined) {
            return false;
        }

        return true;
    }

    function getRgmAreaMarketCoach(coll_id) {
        oRes = new Object();
        oRes.error = 0;
        oRes.errorText = "";
        oRes.result = undefined;

        try {
            var bosses = getBosses(coll_id);
            if (ArrayOptFirstElem(bosses) != undefined) {
                _rgm = ArrayOptFindByKey(bosses, 'RGM', 'level');
                _areaCoach = ArrayOptFindByKey(bosses, 'Area
Coach', 'level');
                _marketCoach = ArrayOptFindByKey(bosses, 'Market
Coach', 'level');
                _rsc = ArrayOptFindByKey(bosses, 'RSC', 'level');

                _result = new Object();
                errCount = 0;
                if (_rgm != undefined) {
                    _result.rgm_id = _rgm.id;
                } else {
                    errCount++;
                }
                if (_areaCoach != undefined) {
                    _result.area_coach_id = _areaCoach.id;
                } else {
                    errCount++;
                }
                if (_marketCoach != undefined) {
                    _result.market_coach_id = _marketCoach.id;
                } else {
                    errCount++;
                }
                if (_rsc != undefined) {
                    _result.rsc_id = _rsc.id;
                } else {
                    errCount++;
                }
                if (errCount != 0) {
                    oRes.error = 1;
                }
            }
        }
    }

```

```

        oRes.errorText = "Ошибка - у Вас не хватает
руководителей для подачи заявки.";
        return oRes;
    }

    } else {
        oRes.error = 1;
        oRes.errorText = "Ошибка - у Вас не хватает
руководителей для подачи заявки.";
        return oRes;
    }
    oRes.result = _result;
    return oRes;

} catch (error) {
    oRes.error = 1;
    oRes.errorText = error;
    return oRes;
}

}

function isNewReqWithNomination(object_id, nomination_id) {
    var sqlReq = " DECLARE @objectId bigint = " + object_id +
"; \
        DECLARE @nominationId bigint = "+ nomination_id + "; \
        SELECT \
        rs.id AS requestId, \
        rs.status_id AS statusId, \
        ccns.code \
        FROM requests rs \
        INNER JOIN request r ON r.id = rs.id \
        INNER JOIN cc_skill_nominations ccns ON ccns.id =
@nominationId \
        INNER JOIN cc_skill_participations lps ON
lps.request_id = rs.id \
        WHERE rs.object_id = @objectId \
        AND lps.nomination_id = @nominationId \
        AND rs.status_id != 'ignore' \
";
    return ArrayOptFirstElem(XQuery("sql: " + sqlReq)) ==
undefined
}

function requestCreate(resp_col_id, collaborator_ids,
nomination_id, team_name, workflow_stage_code) {

    iPersonID = OptInt(resp_col_id);
    nomination_id = OptInt(nomination_id);

    oSourceParam = new Object();
    result = {
        code: 0,
        descr: "",

```

```

        request_id: 0,
    };

    if (resp_col_id == undefined) {
        throw "параметр resp_col_id не является целым числом: " + resp_col_id;
    }

    if (nomination_id == undefined) {
        throw "параметр nomination_id не является целым числом: " + nomination_id;
    }

    if (!isArray(collaborator_ids)) {
        throw "параметр collaborator_ids не является массивом";
    }

    if (workflow_stage_code == undefined) {
        throw "не передан параметр workflow_stage_code";
    }

    if (ArrayCount(collaborator_ids) > 1) {
        for (collaborator_id in collaborator_ids) {
            if (!isBossForCollaborator(collaborator_id, resp_col_id)) {
                throw 'отказано в доступе';
            }
        }
    } else {
        if (OptInt(ArrayOptFirstElem(collaborator_ids)) != OptInt(resp_col_id, 0)) {
            throw 'отказано в доступе';
        }
    }

    getType = ArrayOptFirstElem(XQuery("for $elem in request_types where $elem/code = 'skill' return $elem"));
    if (getType == undefined) {
        throw "не удалось получить идентификатор типа заявки по коду - skill";
    }

    getPeriod = ArrayOptFirstElem(XQuery("sql: SELECT id FROM cc_skill_periods WHERE year = " + Year(Date())));
    if (getPeriod == undefined) {
        throw "не удалось получить идентификатор периода Лиги роста по нынешнему году";
    }

    club_name = "";

    if (ArrayCount(collaborator_ids) > 1) {

```

```

        oSourceParam.is_group = true;
        oSourceParam.SelectedPersons =
ArrayMerge(collaborator_ids, "This", ";");
        club_name = team_name;
    } else {
        oSourceParam.is_group = false;
        club_name =
tools.open_doc(ArrayOptFirstElem(collaborator_ids)).TopElem.fullna
me;
    }

    oSourceParam.selected_object_id =
ArrayOptFirstElem(collaborator_ids);

    if (!oSourceParam.is_group &&
!isNewReqWithNomination(OptInt(oSourceParam.selected_object_id),
OptInt(nomination_id))) {
        result.code = 500;
        result.descr = "Заявка по данной номинации была подана
панее";
        return result;
    }

    forCustomfields =
getRgmAreaMarketCoach(OptInt(oSourceParam.selected_object_id));
    if (forCustomfields.error != 0) {
        result.code = 500;
        result.descr = forCustomfields.errorText;
        return result;
    }

    request_type_id = getType.id;
    requestTypeDoc = tools.open_doc(request_type_id).TopElem;
    tePerson = tools.open_doc(resp_col_id).TopElem;

    oSourceParam.disable_existing_request_check = true;

    oResCreateRequest =
tools.call_code_library_method("libMain", "CreateRequestMain",
[request_type_id, requestTypeDoc, oSourceParam, iPersonID,
tePerson, null]);

    if (oResCreateRequest.error != 0) {
        result.code = 500;
        result.descr = "Ошибка запуска процесса ассессмент: "
+ oResCreateRequest.errorText;
        return result;
    }

    docskillParticip =
tools.new_doc_by_name('cc_skill_participation', false);
    docskillParticip.BindToDb();

```

```

        docskillParticip.TopElem.name = Year(Date()) + " - №" +
oResCreateRequest.doc_request.TopElem.code + " - " + club_name;
        docskillParticip.TopElem.request_id =
oResCreateRequest.doc_request.DocID;
        docskillParticip.TopElem.period_id = getPeriod.id;
        docskillParticip.TopElem.nomination_id = nomination_id;
        docskillParticip.TopElem.rgm_id =
forCustomfields.result.rgm_id;
        docskillParticip.TopElem.area_coach_id =
forCustomfields.result.area_coach_id;
        docskillParticip.TopElem.market_coach_id =
forCustomfields.result.market_coach_id;
        docskillParticip.TopElem.rsc_id =
forCustomfields.result.rsc_id;

        result.request_id = oResCreateRequest.doc_request.DocID;

        if (team_name) {
            docskillParticip.TopElem.team_name = team_name;
        }

        if (oResCreateRequest.doc_request.TopElem.is_group) {
            oResCreateRequest.doc_request.TopElem.workflow_state =
workflow_stage_code == "approving" ? "stage_1" :
workflow_stage_code;
        } else if (ArrayOptFirstElem(XQuery("for $elem in
cc_skill_nominations where $elem/id = " + nomination_id + " return
$elem"), { "code": "" })).code == "party") {
            oResCreateRequest.doc_request.TopElem.workflow_state =
"closed";
            oResCreateRequest.doc_request.TopElem.status_id =
"close";
        } else {
            oResCreateRequest.doc_request.TopElem.workflow_state =
"approving";
        }

        oResCreateRequest.doc_request.Save();

        openReqDoc =
tools.open_doc(oResCreateRequest.doc_request.DocID);
        reqDocTE = openReqDoc.TopElem;
        reqDocTE.custom_elems.ObtainChildByKey('rgm').value =
forCustomfields.result.rgm_id;
        reqDocTE.custom_elems.ObtainChildByKey('area_coach').value
= forCustomfields.result.area_coach_id;

        reqDocTE.custom_elems.ObtainChildByKey('market_coach').value =
forCustomfields.result.market_coach_id;
        reqDocTE.custom_elems.ObtainChildByKey('rsc').value =
forCustomfields.result.rsc_id;
        reqDocTE.custom_elems.ObtainChildByKey('nomination').value
= nomination_id;

```

```

        openReqDoc.Save();

        docskillParticip.Save();

        skill_notifications_tools =
OpenCodeLib(FilePathToUrl(AppDirectoryPath() +
"\custom_tools\custom_skill_notifications_tools.js"));

skill_notifications_tools.congrWithNewStage(OptInt(openReqDoc.DocID))

        return result;
    }

    function requestApprove(request_id) {
        function isWithinDateRange(dateRange) {
            today = new Date();
            return today >= Date(dateRange.start) && today <=
Date(dateRange.end);
        }

        function closeStragglers(request_id, dateRange,
nomination_code) {

            var sqlReq = " \
DECLARE @win_request_id BIGINT = "+ request_id + "; \
\
SELECT rs.workflow_state, \
CASE \
    WHEN rs.workflow_state = 'stage_2' THEN
leag_part.rgm_id \
    WHEN rs.workflow_state = 'stage_3' THEN
leag_part.area_coach_id \
    WHEN rs.workflow_state = 'stage_4' THEN
leag_part.market_coach_id \
    ELSE NULL \
END AS boss_id \
FROM requests rs \
JOIN cc_skill_participations leag_part \
ON leag_part.request_id = rs.id \
INNER JOIN request_types rts \
ON rts.id = rs.request_type_id \
AND rts.code = 'skill' \
WHERE rs.id = @win_request_id \
AND rs.is_group = 0 \
AND rs.workflow_state IN ( 'stage_2', 'stage_3',
'stage_4' ); \
";

            var winReqStagBoss = ArrayOptFirstElem(XQuery("sql: "
+ sqlReq), { workflow_state: "", boss_id: 0 });

```

					ПГУ 1.090501.12.001 ПЗ	Лист
						74
Изм.	Лист	№ докум.	Подпись	Дата		

```

        if ((winReqStagBoss.workflow_state == "stage_2" ||
winReqStagBoss.workflow_state == "stage_3" ||
winReqStagBoss.workflow_state == "stage_4") &&
winReqStagBoss.boss_id != 0) {
            var allReqByBoss =
getRequests(winReqStagBoss.boss_id, dateRange.id);
            var allReqByNomination = ArraySelect(allReqByBoss,
"String(This.nomination_code) == String(nomination_code)");
            var winReqStage =
String(winReqStagBoss.workflow_state).split('_')[1];

            for (req in allReqByNomination) {
                if (OptInt(req.request_id) ==
OptInt(request_id)) continue;
                if (String(req.workflow_state) == "approving"
|| String(req.workflow_state) == "closed") continue;

                var loseReqStage =
String(req.workflow_state).split('_')[1];

                if (loseReqStage < winReqStage) {
                    requestDoc =
tools.open_doc(req.request_id);
                    requestDoc.TopElem.workflow_state =
"closed";

                    requestDoc.TopElem.status_id = "close";
                    requestDoc.Save();
                }
            }
        }

        result = {
            code: 0,
            descr: ""
        };

        status_workflow = ["approving", "stage_1", "stage_2",
"stage_3", "stage_4", "closed"];

        request_id = OptInt(request_id);
        if (request_id == undefined) {
            throw "параметр request_id не является целым числом: "
+ request_id;
        }

        requestDoc = tools.open_doc(request_id);
        if (requestDoc == undefined) {
            throw "заявка с id [ " + request_id + " ] не найдена";
        }

        isGroup = requestDoc.TopElem.is_group;

```

					ПГУ 1.090501.12.001 ПЗ	Лист
						75
Изм.	Лист	№ докум.	Подпись	Дата		

```

        if (requestDoc.TopElem.Name != "request") {
            throw "[ " + request_id + " ] не является документом
заявки";
        }

        nomination_code = ArrayOptFirstElem(XQuery("sql: \
            SELECT leg_noms.code \
            FROM cc_skill_participations leg_pars \
            INNER JOIN cc_skill_nominations leg_noms ON
leg_noms.id = leg_pars.nomination_id \
            WHERE leg_pars.request_id = " + request_id
        ), { code: undefined }).code;

        if (nomination_code == undefined) {
            throw "Отсутствует номинация для заявки [ " +
request_id + " ] ";
        }

        dateRange = ArrayOptFirstElem(XQuery("sql: \
            SELECT * \
            FROM cc_skill_periods leag_per \
            WHERE leag_per.[year] = YEAR(GETDATE()) \
        "));

        if (dateRange == undefined) {
            throw "Отсутствует период лиги роста для текущего года
";
        }

        if (nomination_code == "party" && !isGroup) {
            requestDoc.TopElem.workflow_state =
status_workflow[ArrayCount(status_workflow) - 1];
            requestDoc.TopElem.status_id = "close";

            result.code = 409;
            result.descr = "Заявка является индивидуальной и не
может участвовать в командной номинации [ " + request_id + " ] ";
        } else {
            for (i = 0; i < ArrayCount(status_workflow) - 1; i++)
            {
                if (requestDoc.TopElem.workflow_state ==
status_workflow[i]) {
                    if (i != 0 && !isWithinDateRange({ start:
dateRange.GetOptProperty("stage_" + i + "_start_date"), end:
dateRange.GetOptProperty("stage_" + i + "_finish_date") })) {
                        throw "Нельзя перевести заявку на
следующий этап до завершения текущего этапа";
                    }
                    requestDoc.TopElem.workflow_state =
status_workflow[i + 1];
                    if (ArrayCount(status_workflow) - 2 == i) {
                        requestDoc.TopElem.status_id = "close";
                    }
                }
            }
        }
    }
}

```

```

        break;
    }
}
}
requestDoc.Save();
if (result.code == 0 && nomination_code != "party" &&
!isGroup) {
    closeStragglers(requestDoc.DocID, dateRange,
nomination_code)
}

    skill_notifications_tools =
OpenCodeLib(FilePathToUrl(AppDirectoryPath() +
"\custom_tools\custom_skill_notifications_tools.js"));

skill_notifications_tools.congrWithNewStage(OptInt(requestDoc.DocI
D))

    return result;
}

function requestCancel(request_id, cur_user_id) {
    request_id = OptInt(request_id);
    if (request_id == undefined) {
        throw "параметр request_id не является целым числом: "
+ request_id;
    }

    requestDoc = tools.open_doc(request_id);
    if (requestDoc == undefined) {
        throw "заявка с id [ " + request_id + " ] не найдена";
    }

    requestParticipation = ArrayOptFirstElem(XQuery("sql:
SELECT * FROM cc_skill_participations WHERE request_id = " +
request_id), {
        rgm_id: undefined,
        area_coach_id: undefined,
        market_coach_id: undefined
    });

    if (requestParticipation.rgm_id != cur_user_id
        && requestParticipation.area_coach_id != cur_user_id
        && requestParticipation.market_coach_id != cur_user_id
    ) {
        throw "отказано в доступе";
    }

    requestDoc.TopElem.status_id = "ignore";
    requestDoc.Save();
}

function getCountRequestsByCurPeriod() {

```

```

year = Year(Date());

sqlReq = " DECLARE @year int = " + year + "; \
        SELECT \
            COUNT(rs.id) AS count_req \
        FROM requests rs \
            INNER JOIN cc_skill_participations cclp ON
cclp.request_id = rs.id \
            INNER JOIN cc_skill_periods cclpd ON cclpd.id =
cclp.period_id AND cclpd.[year] = @year \
        ";

return ArrayOptFirstElem(XQuery("sql: " + sqlReq), {
count_req: 0 });
}

function getRequests(boss_id, period_id) {
    boss_id = OptInt(boss_id);
    period_id = OptInt(period_id);

    if (boss_id == undefined) {
        throw "значение параметра boss_id, задано некорректно:
" + boss_id;
    }

    if (period_id == undefined) {
        getPeriod = ArrayOptFirstElem(XQuery("sql: SELECT id
FROM cc_skill_periods WHERE year = " + Year(Date())));

        if (getPeriod == undefined) {
            throw "не удалось получить идентификатор периода
Лиги роста по нынешнему году";
        }
        period_id = getPeriod.id;
    }

    sql =
        "sql: \
        DECLARE @coll_id bigint = " + boss_id + "; \
        DECLARE @period_id BIGINT = " + period_id + "; \
        \
        WITH main_data AS ( \
            SELECT \
                rs.id AS request_id, \
                rs.is_group, \
                rs.workflow_state, \
                rs.status_id, \
                rs.object_id, \
                leag_part.name, \
                leag_part.team_name, \
                nom.name AS nomination_name, \
                nom.code AS nomination_code, \
                leag_part.rgm_id, \

```

```

        leag_part.area_coach_id, \
        leag_part.market_coach_id, \
        CASE \
            WHEN rs.is_group = 0 AND nom.code =
'party' THEN CAST(1 AS BIT) \
            ELSE CAST(0 AS BIT) \
        END AS is_join_to_team \
    FROM requests rs \
    JOIN cc_skill_participations leag_part ON
leag_part.request_id = rs.id \
    INNER JOIN request_types rts ON rts.id =
rs.request_type_id AND rts.code = 'skill' \
    INNER JOIN cc_skill_nominations nom ON nom.id =
leag_part.nomination_id \
    JOIN request r ON r.id = rs.id \
    WHERE leag_part.period_id = @period_id \
), \
Participants AS ( \
    SELECT \
        md.request_id, \
        c.id AS participant_id, \
        c.fullname, \
        c.position_name, \
        c.position_parent_name \
    FROM main_data md \
    JOIN request req ON req.id = md.request_id AND
md.is_group = 1 \
    CROSS APPLY
req.data.nodes('/request/persons/person') AS cl(data) \
    JOIN collaborators c ON c.id =
cl.data.value('(person_id)[1]', 'bigint') \
    \
    UNION ALL \
    \
    SELECT \
        md.request_id, \
        md.object_id AS participant_id, \
        c.fullname, \
        c.position_name, \
        c.position_parent_name \
    FROM main_data md \
    JOIN collaborators c ON c.id = md.object_id \
    WHERE md.is_group = 0 \
), \
AssessmentsMapped AS ( \
    SELECT \
        la.request_id, \
        as_aps.name, \
        as_aps.start_date, \
        as_aps.status, \
        cls.data.value('(collaborator/firstname)[1]',
'nvarchar(max)') + ' ' + \

```

```

cls.data.value('(collaborator/lastname)[1]', 'nvarchar(max)') AS
expert_name, \
                                part.value('(role)[1]', 'nvarchar(max)') AS
role, \

type_elem.value('(assessment_appraise_type_id)[1]',
'nvarchar(50)') AS appraise_type_id, \
                                pa.data.value('(pa/overall)[1]',
'nvarchar(max)') AS overall, \
                                la.workflow_state, \
                                '/appr_player.html?assessment_appraise_id=' +
CAST(as_aps.id AS NVARCHAR) + '&pa_id=' + CAST(pa.id AS NVARCHAR)
AS link \
                                FROM cc_skill_apprs la \
                                INNER JOIN assessment_appraise as_ap ON
la.assessment_appraise_id = as_ap.id \
                                INNER JOIN assessment_appraises as_aps ON
as_aps.id = as_ap.id \
                                CROSS APPLY
as_ap.data.nodes('/assessment_appraise') AS app(data) \
                                CROSS APPLY
app.data.nodes('participants/participant/customize/experts/expert'
) AS part(part) \
                                CROSS APPLY
app.data.nodes('participants/participant/assessment_appraise_types
/assessment_appraise_type') AS type_elem(type_elem) \
                                LEFT JOIN pa ON
TRY_CAST(pa.data.value('(pa/assessment_appraise_id)[1]',
'nvarchar(50)') AS BIGINT) = as_ap.id \
                                LEFT JOIN collaborator cls ON cls.id =
part.value('(person_id)[1]', 'nvarchar(max)') \
                                ) \
                                \
                                SELECT \
                                md.request_id, \
                                md.is_group, \
                                md.workflow_state, \
                                md.status_id, \
                                md.object_id, \
                                md.name, \
                                md.team_name, \
                                md.nomination_name, \
                                md.nomination_code, \
                                md.is_join_to_team, \
                                CASE \
                                WHEN md.rgm_id = @coll_id THEN 'rgm' \
                                WHEN md.area_coach_id = @coll_id THEN
'area_coach' \
                                WHEN md.market_coach_id = @coll_id THEN
'market_coach' \
                                ELSE NULL \
                                END AS type, \

```

					ПГУ 1.090501.12.001 ПЗ	Лист
						80
Изм.	Лист	№ докум.	Подпись	Дата		

```

ISNULL(( \
    SELECT \
        participant_id, \
        fullname, \
        position_name, \
        position_parent_name \
    FROM Participants \
    WHERE request_id = md.request_id \
    FOR JSON PATH, INCLUDE_NULL_VALUES \
), '[]') AS participant_info, \
ISNULL(( \
    SELECT \
        name, \
        start_date, \
        status, \
        expert_name, \
        role, \
        appraise_type_id, \
        overall, \
        workflow_state, \
        link \
    FROM AssessmentsMapped \
    WHERE request_id = md.request_id \
    FOR JSON PATH, INCLUDE_NULL_VALUES \
), '[]') AS assessment_info \
FROM main_data md \
WHERE \
    md.rgm_id = @coll_id OR \
    md.area_coach_id = @coll_id OR \
    md.market_coach_id = @coll_id OR \
    md.rsc_id = @coll_id; \
";

```

```

allData = ArraySelectAll(XQuery(sql));
result = new Array();
currentRequest = null;
if (ArrayOptFirstElem(allData) == undefined) {
    return result;
}

for (item in allData) {
    result.push({
        request_id: RValue(item.request_id),
        is_group: RValue(item.is_group),
        workflow_state: RValue(item.workflow_state),
        status_id: RValue(item.status_id),
        name: RValue(item.name),
        team_name: RValue(item.team_name),
        nomination_name: RValue(item.nomination_name),
        nomination_code: RValue(item.nomination_code),
        is_join_to_team: RValue(item.is_join_to_team),
        participant_info:
tools.read_object(RValue(item.participant_info)),

```

```

        assessment_info:
tools.read_object(RValue(item.assessment_info)),
    });
}

if (currentRequest != null) {
    result.push(currentRequest);
}

return result;
}

function getRequestsForTournamentTable(boss_id, period_id) {
    boss_id = OptInt(boss_id);
    period_id = OptInt(period_id);

    if (boss_id == undefined) {
        throw "значение параметра boss_id, задано некорректно: "
+ boss_id;
    }

    if (period_id == undefined) {
        getPeriod = ArrayOptFirstElem(XQuery("sql: SELECT id
FROM cc_skill_periods WHERE year = " + Year(Date())));

        if (getPeriod == undefined) {
            throw "не удалось получить идентификатор периода Лиги
роста по нынешнему году";
        }
        period_id = getPeriod.id;
    }

    sql =
        "sql: \
        DECLARE @coll_id bigint = " + boss_id + "; \
        DECLARE @period_id BIGINT = " + period_id + "; \
        \
        WITH main_data AS ( \
            SELECT \
                rs.id AS request_id, \
                rs.is_group, \
                rs.workflow_state, \
                rs.status_id, \
                rs.object_id, \
                leag_part.name, \
                leag_part.team_name, \
                nom.name AS nomination_name, \
                nom.code AS nomination_code, \
                leag_part.rgm_id, \
                leag_part.area_coach_id, \
                leag_part.market_coach_id, \
                CASE \

```

					ПГУ 1.090501.12.001 ПЗ	Лист
						82
Изм.	Лист	№ докум.	Подпись	Дата		

```

        WHEN rs.is_group = 0 AND nom.code =
'party' THEN CAST(1 AS BIT) \
        ELSE CAST(0 AS BIT) \
        END AS is_join_to_team \
FROM requests rs \
JOIN cc_skill_participations leag_part ON
leag_part.request_id = rs.id \
INNER JOIN request_types rts ON rts.id =
rs.request_type_id AND rts.code = 'skill' \
INNER JOIN cc_skill_nominations nom ON nom.id =
leag_part.nomination_id \
JOIN request r ON r.id = rs.id \
WHERE leag_part.period_id = @period_id \
), \
Participants AS ( \
SELECT \
    md.request_id, \
    c.id AS participant_id, \
    c.fullname, \
    c.position_name, \
    c.position_parent_name, \
    (SELECT TOP(1) \
        st.subdivision_id, \
        st.subdivision_name, \
        st.parent_object_id, \
        (select TOP 1 sub.name from \
            subdivisions_tree s \
            JOIN subs sub on sub.id =
s.parent_object_id \
                WHERE s.subdivision_id =
st.subdivision_id AND s.level = MAX(st.level)-1) as
partner_franchise_name, \
        ss.name as top_level_name, \
        MAX(st.level) as level \
        from subdivisions_tree st \
        JOIN collaborators cs on
cs.position_parent_id = st.subdivision_id \
        JOIN subs ss on ss.id =
st.parent_object_id \
            WHERE cs.id = c.id \
        GROUP BY \
            st.subdivision_id, \
            st.subdivision_name, \
            st.parent_object_id, \
            ss.name \
        ORDER BY level DESC \
        FOR JSON PATH, WITHOUT_ARRAY_WRAPPER)
as partners \
FROM main_data md \
JOIN request req ON req.id = md.request_id AND
md.is_group = 1 \
CROSS APPLY
req.data.nodes('/request/persons/person') AS cl(data) \

```

```

        JOIN collaborators c ON c.id =
cl.data.value('(person_id)[1]', 'bigint') \
    \
    UNION ALL \
    \
    SELECT \
        md.request_id, \
        md.object_id AS participant_id, \
        c.fullname, \
        c.position_name, \
        c.position_parent_name, \
        (SELECT TOP(1) \
            st.subdivision_id, \
            st.subdivision_name, \
            st.parent_object_id, \
            (select TOP 1 sub.name from \
                subdivisions_tree s \
                JOIN subs sub on sub.id =
s.parent_object_id \
                                WHERE s.subdivision_id =
st.subdivision_id AND s.level = MAX(st.level)-1) as
partner_franchise_name, \
            ss.name as top_level_name, \
            MAX(st.level) as level \
            from subdivisions_tree st \
            JOIN collaborators cs on
cs.position_parent_id = st.subdivision_id \
            JOIN subs ss on ss.id =
st.parent_object_id \
                                WHERE cs.id = c.id \
        GROUP BY \
            st.subdivision_id, \
            st.subdivision_name, \
            st.parent_object_id, \
            ss.name \
        ORDER BY level DESC \
        FOR JSON PATH, WITHOUT_ARRAY_WRAPPER)
as partners \
        FROM main_data md \
        JOIN collaborators c ON c.id = md.object_id \
        WHERE md.is_group = 0 \
    ), \
    AssessmentsMapped AS ( \
        SELECT \
            la.request_id, \
            as_aps.name, \
            as_aps.start_date, \
            as_aps.status, \
            cls.data.value('(collaborator/firstname)[1]',
'nvarchar(max)') + ' ' + \
            cls.data.value('(collaborator/lastname)[1]', 'nvarchar(max)') AS
expert_name, \

```

```

part.value('(role)[1]', 'nvarchar(max)') AS
role, \

type_elem.value('(assessment_appraise_type_id)[1]',
'nvarchar(50)') AS appraise_type_id, \
pa.data.value('(pa/overall)[1]',
'nvarchar(max)') AS overall, \
la.workflow_state, \
'/appr_player.html?assessment_appraise_id=' +
CAST(as_aps.id AS NVARCHAR) + '&pa_id=' + CAST(pa.id AS NVARCHAR)
AS link \

FROM cc_skill_apprs la \
INNER JOIN assessment_appraise as_ap ON
la.assessment_appraise_id = as_ap.id \
INNER JOIN assessment_appraises as_aps ON
as_aps.id = as_ap.id \
CROSS APPLY
as_ap.data.nodes('/assessment_appraise') AS app(data) \
CROSS APPLY
app.data.nodes('participants/participant/customize/experts/expert'
) AS part(part) \
CROSS APPLY
app.data.nodes('participants/participant/assessment_appraise_types
/assessment_appraise_type') AS type_elem(type_elem) \
LEFT JOIN pa ON
TRY_CAST(pa.data.value('(pa/assessment_appraise_id)[1]',
'nvarchar(50)') AS BIGINT) = as_ap.id \
LEFT JOIN collaborator cls ON cls.id =
part.value('(person_id)[1]', 'nvarchar(max)') \
) \
\
SELECT \
md.request_id, \
md.is_group, \
md.workflow_state, \
md.status_id, \
md.object_id, \
md.name, \
md.team_name, \
md.nomination_name, \
md.nomination_code, \
md.is_join_to_team, \
CASE \
WHEN md.rgm_id = @coll_id THEN 'rgm' \
WHEN md.area_coach_id = @coll_id THEN
'area_coach' \
WHEN md.market_coach_id = @coll_id THEN
'market_coach' \
ELSE NULL \
END AS type, \
ISNULL(( \
SELECT \
p.participant_id, \

```

```

        p.fullname, \
        p.position_name, \
        p.position_parent_name, \
        JSON_QUERY(p.partners) as partners \
    FROM Participants p \
    WHERE request_id = md.request_id \
    FOR JSON PATH, INCLUDE_NULL_VALUES \
), '[]') AS participant_info, \
ISNULL(( \
    SELECT \
        name, \
        start_date, \
        status, \
        expert_name, \
        role, \
        appraise_type_id, \
        overall, \
        workflow_state, \
        link \
    FROM AssessmentsMapped \
    WHERE request_id = md.request_id \
    FOR JSON PATH, INCLUDE_NULL_VALUES \
), '[]') AS assessment_info \
FROM main_data md; \
";

```

```

allData = ArraySelectAll(XQuery(sql));
result = new Array();
currentRequest = null;
if (ArrayOptFirstElem(allData) == undefined) {
    return result;
}

```

```

for (item in allData) {

```

```

    result.push({
        request_id: RValue(item.request_id),
        is_group: RValue(item.is_group),
        workflow_state: RValue(item.workflow_state),
        status_id: RValue(item.status_id),
        name: RValue(item.name),
        team_name: RValue(item.team_name),
        nomination_name: RValue(item.nomination_name),
        nomination_code: RValue(item.nomination_code),
        is_join_to_team: RValue(item.is_join_to_team),
        participant_info:
tools.read_object(RValue(item.participant_info)),
        assessment_info:
tools.read_object(RValue(item.assessment_info)),
    });
}

```

```

if (currentRequest != null) {

```

```

        result.push(currentRequest);
    }

    return result;
}

function getRequestsByCollaborator(collaborator_id, period_id)
{
    collaborator_id = OptInt(collaborator_id);
    period_id = OptInt(period_id);

    if (collaborator_id == undefined) {
        throw "значение параметра collaborator_id, задано
некорректно: " + collaborator_id;
    }

    if (period_id == undefined) {
        getPeriod = ArrayOptFirstElem(XQuery("sql: SELECT id
FROM cc_skill_periods WHERE year = " + Year(Date())));
        if (getPeriod == undefined) {
            throw "не удалось получить идентификатор периода
Лиги роста по нынешнему году";
        }

        period_id = getPeriod.id;
    }

    req = "sql: \
    DECLARE @cur_user_id BIGINT = " + collaborator_id + ";
\
    DECLARE @period_id BIGINT = " + period_id + "; \
    DECLARE @skill_type_id BIGINT; \
\
    SELECT @skill_type_id = id \
    FROM request_types \
    WHERE code = 'skill'; \
\
    WITH raw_requests AS (\
        SELECT \
            reqs.id AS request_id, \
            reqs.is_group, \
            reqs.workflow_state, \
            reqs.status_id, \
            leag_part.name, \
            clns.name AS nomination_name, \
            clns.id AS nomination_id, \
            clns.code AS nomination_code \
        FROM cc_skill_participations leag_part \
        INNER JOIN requests reqs ON reqs.id =
leag_part.request_id AND reqs.request_type_id = @skill_type_id \
        INNER JOIN request req ON req.id =
leag_part.request_id \

```

```

        LEFT JOIN cc_skill_nominations clns ON clns.id =
leag_part.nomination_id \
        WHERE leag_part.period_id = @period_id \
        AND( \
            (reqs.is_group = 0 AND reqs.object_id =
@cur_user_id) \
            OR \
            (reqs.is_group = 1 AND EXISTS( \
                SELECT 1 \
                FROM request r \
                CROSS APPLY
r.data.nodes('/request/persons/person') AS cl(data) \
                WHERE cl.data.value('(person_id)[1]',
'bigint') = @cur_user_id \
                    AND r.id = reqs.id \
            )) \
        )) select * from raw_requests\
";

return XQuery(req);
}

function getAvailableCollaboratorsForTeam(boss_id, request_id,
workflow_stage_code) {
    boss_id = OptInt(boss_id);
    request_id = OptInt(request_id);

    if (boss_id == undefined) {
        throw "значение параметра boss_id, задано некорректно:
" + boss_id;
    }

    if (workflow_stage_code == undefined) {
        throw "значение параметра workflow_stage_code, задано
некорректно: " + workflow_stage_code;
    }

    if (request_id == undefined) {
        request_id = "NULL";
    }

    getPeriod = ArrayOptFirstElem(XQuery("sql: SELECT id FROM
cc_skill_periods WHERE year = " + Year(Date())));
    if (getPeriod == undefined) {
        throw "не удалось получить идентификатор периода Лиги
роста по нынешнему году";
    }

    period_id = getPeriod.id;

    req = "sql: \
        DECLARE @current_user_id BIGINT = " + boss_id + "; \

```

					ПГУ 1.090501.12.001 ПЗ	Лист
						88
Изм.	Лист	№ докум.	Подпись	Дата		

```

        DECLARE @target_period_id BIGINT = " + period_id + ";
\
        DECLARE @target_request_id BIGINT = " + request_id +
"; \
        DECLARE @global_workflow_stage NVARCHAR(max) = " +
workflow_stage_code + "; \
        DECLARE @skill_request_type_id BIGINT; \
\
        SELECT @skill_request_type_id = id \
        FROM request_types \
        WHERE code = 'skill'; \
\
        -- Этапы ДО по порядку \
        DECLARE @WorkflowStates TABLE ( \
            state_id NVARCHAR(50), \
            position INT \
        ); \
\
        INSERT INTO @WorkflowStates \
        VALUES \
            ('approving', 0), \
            ('stage_1', 1), \
            ('stage_2', 2), \
            ('stage_3', 3), \
            ('stage_4', 4), \
            ('closed', 5); \
\
        WITH \
            SubordinateEmployees AS ( \
                SELECT id AS person_id \
                FROM
get_sub_person_ids_by_func_manager_id(@current_user_id) \
            ), \
\
            -- Сотрудники с одобренными групповыми заявками на
номинацию 'party' \
            ApprovedPartyParticipants AS ( \
                SELECT DISTINCT r.object_id AS person_id \
                FROM requests r \
                JOIN cc_skill_participations lp ON
lp.request_id = r.id \
                JOIN cc_skill_nominations ln ON ln.id =
lp.nomination_id \
                WHERE r.request_type_id =
@skill_request_type_id \
                AND lp.period_id = @target_period_id \
                AND r.status_id = 'close' \
                AND ln.code = 'party' \
                AND r.is_group = 0 \
            ), \
\
            -- Все групповые заявки \
            AllGroupRequests AS ( \

```

```

SELECT \
    r.id AS request_id, \
    r.status_id, \
    r.workflow_state, \
    lp.period_id \
FROM requests r \
JOIN cc_skill_participations lp ON
lp.request_id = r.id \
    WHERE r.request_type_id =
@skill_request_type_id \
    AND lp.period_id = @target_period_id \
    AND r.is_group = 1 \
), \
\
-- Сотрудники из активных групповых заявок \
EmployeesInGroupRequests AS ( \
    SELECT DISTINCT \
        x.data.value('(person_id)[1]', 'BIGINT')
AS person_id \
        FROM request req \
        JOIN requests r ON r.id = req.id \
        JOIN AllGroupRequests agrs ON agrs.request_id
= req.id \
        CROSS APPLY
req.data.nodes('/request/persons/person') AS x(data) \
        WHERE agrs.status_id <> 'ignore' \
), \
\
-- Сотрудники из отклоненных заявок для повторной
подачи \
RejectedRequestCandidates AS ( \
    SELECT DISTINCT \
        x.data.value('(person_id)[1]', 'BIGINT')
AS person_id \
        FROM request req \
        JOIN requests r ON r.id = req.id \
        JOIN AllGroupRequests agr ON agr.request_id =
req.id \
        CROSS APPLY
req.data.nodes('/request/persons/person') AS x(data) \
        JOIN @WorkflowStates gwks ON gwks.state_id =
@global_workflow_stage \
        JOIN @WorkflowStates agr_ws ON
agr_ws.state_id = agr.workflow_state \
        WHERE r.status_id = 'ignore' \
        AND ( \
            (gwks.position >= 1 AND
(agr_ws.position = gwks.position - 1 OR agr_ws.position =
gwks.position)) \
            OR \
            (gwks.position = 0 AND agr_ws.position
= 0) \
        ) \

```

```

        ) \
    \
    SELECT \
        se.person_id AS id, \
        c.fullname, \
        c.position_name \
    FROM SubordinateEmployees se \
    LEFT JOIN requests trg ON trg.id = @target_request_id
\
        LEFT JOIN @WorkflowStates wkss ON wkss.state_id =
trg.workflow_state \
        JOIN @WorkflowStates gwks ON gwks.state_id =
@global_workflow_stage \
        JOIN collaborators c ON c.id = se.person_id \
    WHERE \
        ( \
            ( \
                -- Создание новой команды \
                @target_request_id IS NULL AND \
                ( \
                    -- Этап регистрации \
                    gwks.position = 0 \
                    OR \
                    -- Повторная подача отклоненной заявки
\
                        EXISTS ( \
                            SELECT 1 FROM
RejectedRequestCandidates rrc \
                                WHERE rrc.person_id = se.person_id
\
                                    ) \
                                ) \
                            ) OR ( \
                                -- Редактирование команды \
                                @target_request_id IS NOT NULL AND \
                                ( \
                                    -- Этап регистрации \
                                    gwks.position = 0 \
                                    OR \
                                    -- Повторная подача отклоненной заявки
\
                                        EXISTS ( \
                                            SELECT 1 FROM
RejectedRequestCandidates rrc \
                                                WHERE rrc.person_id = se.person_id
\
                                                    ) \
                                                ) \
                                            ) \
                                        ) AND \
                                        -- Индивидуальные, одобренные в групповой
номинации \
                                            EXISTS \

```

					ПГУ 1.090501.12.001 ПЗ	Лист
						91
Изм.	Лист	№ докум.	Подпись	Дата		

```

        ( \
            SELECT 1 \
            FROM ApprovedPartyParticipants app \
            WHERE app.person_id = se.person_id \
        ) \
        -- Нет активных или одобренных групповых заявок \
        AND NOT EXISTS ( \
            SELECT 1 FROM EmployeesInGroupRequests eg \
            WHERE eg.person_id = se.person_id \
        ) \
        ORDER BY c.fullname; \
";

return XQuery(req);
}

function getPermissionRules(user_id) {
    user_id = OptInt(user_id);

    if (user_id == undefined) {
        throw "не передан обязательный параметр user_id";
    }

    function isWithinDateRange(dateRange) {
        today = new Date();
        return today >= Date(dateRange.start) && today <=
Date(dateRange.end);
    }

    result = {
        approving: {
            request_viewing: true,
            request_management: false,
            team_management: false,
        },
        stage_1: {
            request_create: false,
            request_viewing: false,
            request_management: false,
            team_management: false,
            checklist_viewing: false,
            checklist_management: false,
            checklist_viewing_ext: false,
            move_next: false,
            request_reject: false
        },
        stage_2: {
            checklist_viewing: false,
            checklist_management: false,
            checklist_viewing_ext: false,
            team_management: false,
            move_next: false,
            request_reject: false
        }
    }
}

```

```

    },
    stage_3: {
        checklist_viewing: false,
        checklist_management: false,
        checkist_viewing_ext: false,
        team_management: false,
        move_next: false,
        request_reject: false
    },
    stage_4: {
        checklist_viewing: false,
        checklist_management: false,
        checkist_viewing_ext: false,
        team_management: false,
        move_next: false,
        request_reject: false
    }
};

coll_info = ArrayOptFirstElem(XQuery("sql: SELECT * FROM
collaborators WHERE position_name = 'RGM' and id = " + user_id));

if (coll_info != undefined) {
    result.approving.request_management = true;
    result.approving.team_management = true;
}

sql_person_type = "sql: \
DECLARE @coll_id bigint = "+ user_id + "; \
with \
    main_data as ( \
        SELECT \
            rs.id as request_id, \
            rs.status_id, \
            lps.rgm_id, \
            lps.area_coach_id, \
            lps.market_coach_id, \
            lps.rsc_id \
        FROM requests rs \
        INNER JOIN request_types rts ON rts.id =
rs.request_type_id AND rts.code = 'skill' \
        INNER JOIN request r on r.id = rs.id \
        INNER JOIN cc_skill_participations lps ON
lps.request_id = rs.id \
    ) \
    SELECT TOP(1) \
        CASE \
            WHEN rgm_id = @coll_id THEN 'rgm' \
            WHEN area_coach_id = @coll_id THEN
'area_coach' \
            WHEN market_coach_id = @coll_id THEN
'market_coach' \
            WHEN rsc_id = @coll_id THEN 'rsc' \

```

```

        ELSE NULL \
        END as type \
    FROM main_data \
    where(rgm_id = @coll_id OR area_coach_id = @coll_id OR
market_coach_id = @coll_id OR rsc_id = @coll_id) AND status_id =
'active' \
    ";

    dateRange = ArrayOptFirstElem(XQuery("sql: \
    SELECT * \
    FROM cc_skill_periods leag_per \
    WHERE leag_per.[year] = YEAR(GETDATE()) \
    "));

    user_type_request =
ArrayOptFirstElem(XQuery(sql_person_type))
    if (user_type_request != undefined && dateRange !=
undefined) {
        isStage1 = isWithinDateRange({ start:
dateRange.stage_1_start_date, end: dateRange.stage_1_finish_date
});

        isStage2 = isWithinDateRange({ start:
dateRange.stage_2_start_date, end: dateRange.stage_2_finish_date
});

        isStage3 = isWithinDateRange({ start:
dateRange.stage_3_start_date, end: dateRange.stage_3_finish_date
});

        result.stage_1.request_viewing = true;
        result.stage_2.request_viewing = true;
        result.stage_3.request_viewing = true;

        switch (user_type_request.type) {
            case 'rgm':
                if (isStage1) {
                    result.stage_1.request_create = false;
                    result.stage_1.request_management = true;
                    result.stage_1.team_management = true;
                    result.stage_1.checklist_viewing = true;
                    result.stage_1.checklist_management =
true;

                    result.stage_1.checkist_viewing_ext =
true;

                    result.stage_1.move_next = true;
                    result.stage_1.request_reject = true;
                }
                if (isStage2) {
                    result.stage_2.checklist_viewing = true;
                    result.stage_2.checklist_management =
true;

                    result.stage_2.checkist_viewing_ext =
false;
                }
            }
        }
    }

```

```

        break;
    case 'area_coach':
        result.stage_1.checklist_viewing = true;
        result.stage_1.checkist_viewing_ext = true;
        if (isStage2) {
            result.stage_2.checklist_viewing = true;
            result.stage_2.checklist_management =
true;

            result.stage_2.checkist_viewing_ext =
false;

            result.stage_2.team_management = true;
            result.stage_2.move_next = true;
            result.stage_2.request_reject = true;
        }
        if (isStage3) {
            result.stage_3.checklist_viewing = true;
            result.stage_3.checklist_management =
true;

            result.stage_3.checkist_viewing_ext =
false;

        }
        break;
    case 'market_coach':
        result.stage_1.checklist_viewing = true;
        result.stage_1.checkist_viewing_ext = true;
        result.stage_2.checklist_viewing = true;
        result.stage_2.checkist_viewing_ext = false;
        if (isStage3) {
            result.stage_3.checklist_viewing = true;
            result.stage_3.checklist_management =
true;

            result.stage_3.checkist_viewing_ext =
false;

            result.stage_3.team_management = true;
            result.stage_3.move_next = true;
            result.stage_3.request_reject = true;
        }
        break;
    case 'rsc':
        result.stage_1.checklist_viewing = true;
        result.stage_1.checkist_viewing_ext = true;
        result.stage_2.checklist_viewing = true;
        result.stage_2.checkist_viewing_ext = false;
        if (isStage3) {
            result.stage_3.checklist_viewing = true;
            result.stage_3.checklist_management =
true;

            result.stage_3.checkist_viewing_ext =
false;

        }
    }
}

```

```

        isStage4 = isWithinDateRange({ start:
dateRange.stage_4_start_date, end: dateRange.stage_4_finish_date
});

        existJudgmentGroup = ArrayOptFirstElem(XQuery("sql: SELECT
id FROM group_collaborators WHERE code = 'skill_judges' AND
collaborator_id = " + user_id)) != undefined;

        if (existJudgmentGroup && isStage4) {
            result.stage_4.checklist_viewing = true;
            result.stage_4.checklist_management = true;
            result.stage_4.checkist_viewing_ext = true;
            result.stage_4.move_next = true;
        }

        return result;
    }

    function updateRequestTeam(cur_user_id, collaborator_ids,
request_id, team_name) {
        result = {
            code: 0,
            descr: ""
        };

        request_id = OptInt(request_id);
        if (request_id == undefined) {
            throw "значение параметра request_id, задано
некорректно: " + request_id;
        }

        if (!isArray(collaborator_ids)) {
            throw "параметр collaborators не является массивом";
        }

        if (ArrayCount(collaborator_ids) > 1) {
            for (collaborator_id in collaborator_ids) {
                if (!isBossForCollaborator(collaborator_id,
cur_user_id)) {
                    throw 'отказано в доступе';
                }
            }
        } else {
            if (OptInt(ArrayOptFirstElem(collaborator_ids)) !=
OptInt(cur_user_id, 0)) {
                throw 'отказано в доступе';
            }
        }

        reqDoc = tools.open_doc(request_id);
        reqDocTE = reqDoc.TopElem;
    }

```

```

        requestParticipation = ArrayOptFirstElem(XQuery("sql:
SELECT * FROM cc_skill_participations WHERE request_id = " +
request_id), {
    rgm_id: undefined,
    area_coach_id: undefined,
    market_coach_id: undefined
});

if (requestParticipation.rgm_id != cur_user_id
    && requestParticipation.area_coach_id != cur_user_id
    && requestParticipation.market_coach_id != cur_user_id
) {
    throw "отказано в доступе";
}

reqDocTE.persons.Clear();

for (col_id in collaborator_ids) {
    col_id = OptInt(col_id);
    if (col_id == undefined) {
        result.code = 400;
        result.descr += "col_id [ " + col_id + " ] не
является числом; \n"
        continue;
    }

    person_doc = reqDocTE.persons.AddChild();
    person_doc.person_id = col_id;
    tools.common_filling("collaborator", person_doc,
col_id)
}

if (team_name) {

    docskillParticipId = ArrayOptFirstElem(XQuery("sql:
SELECT id FROM cc_skill_participations WHERE request_id = " +
request_id));

    if (docskillParticipId != undefined) {
        var docskillParticip =
tools.open_doc(docskillParticipId.id);
        var docskillParticipTE = docskillParticip.TopElem;

        docskillParticipTE.team_name = team_name;

        var currentNameSplited =
String(docskillParticipTE.name).split(" - ");

        if (ArrayCount(currentNameSplited) == 3) {
            docskillParticipTE.name =
currentNameSplited[0] + " - " + currentNameSplited[1] + " - " +
team_name;

```

```

        docskillParticip.Save();
    }
}

reqDoc.Save();

return result;
}

function getNominations() {
    req = "sql: \
        SELECT id, code, name \
        FROM cc_skill_nominations \
        "

    return XQuery(req);
}

function getWorkflowStages() {
    req = "sql: \
        SELECT id \
        FROM workflows \
        WHERE code = 'skill' \
        ";

    workflow_id = ArrayOptFirstElem(XQuery(req), { id:
undefined }).id;
    if (workflow_id == undefined) {
        throw "Отсутствует ДО для Лиги роста";
    }

    listWorkflow = [];

    workflowDoc = tools.open_doc(workflow_id);
    workflowDocTE = workflowDoc.TopElem;
    for (i = 0; i < ArrayCount(workflowDocTE.states); i++) {
        workflow_state = workflowDocTE.states[i];
        listWorkflow[i] = {
            code: RValue(workflow_state.code),
            name: RValue(workflow_state.name)
        }
    }

    return listWorkflow;
}

function getRequestStatuses() {
    req = "sql: \
        SELECT id, name \
        FROM[common.request_status_types] \
        ";

```

```

        return XQuery(req);
    }

    function createAssessmentAppraiseById(etalon_aap_id,
auditor_user_id, evaluate_user_id) {
        oRes = new Object();
        oRes.error = 0;
        oRes.errorText = "";
        oRes.result = undefined;

        checkIncomingData = '';
        if (ArrayOptFirstElem(XQuery("for $elem in
assessment_appraises where $elem/id = " + OptInt(etalon_aap_id) +
" return $elem"))) == undefined) {
            checkIncomingData += 'Не найдена эталонная процедура
оценки.'
        }
        _auditor = tools.open_doc(OptInt(auditor_user_id));
        if (_auditor != undefined) {
            var auditorTE = _auditor.TopElem;
        } else {
            checkIncomingData += (checkIncomingData != '' ? ' ' :
'') + 'Не найден эксперт.';
        }
        _evaluate = tools.open_doc(OptInt(evaluate_user_id));
        if (_evaluate != undefined) {
            var evaluateTE = _evaluate.TopElem;
        } else {
            checkIncomingData += (checkIncomingData != '' ? ' ' :
'') + 'Не найден оцениваемый.';
        }

        if (checkIncomingData != '') {
            oRes.error = 1;
            oRes.errorText = checkIncomingData;
            return oRes
        }

        try {
            personDoc =
OpenDoc (UrlFromDocID (OptInt (evaluate_user_id, 0)));
            personTE = personDoc.TopElem;

            new_AA = tools.new_doc_by_name("assessment_appraise",
false);
            new_AA.BindToDb();
            etalon_id = OptInt(etalon_aap_id);
            etalon = OpenDoc (UrlFromDocID(etalon_id));
            etalon_TE = etalon.TopElem
            new_AA_TE = new_AA.TopElem;
            new_AA_TE.AssignElem(etalon_TE);
            new_AA_TE.name += ' ' + personTE.fullname + " " +
CurDate;

```

```

        new_AA_TE.is_model = false;
        new_AA_TE.start_date = Date();
        new_AA_TE.status = '0';
        new_AA_TE.person_id = auditorTE.id;
        auditor = new_AA_TE.auditorys.AddChild();
        auditor.person_id = personTE.id;
        auditor.person_name = personTE.fullname;
        auditor.position_name = personTE.position_name;

        _chExperts =
ArrayOptFirstElem(new_AA_TE.participants).customize.experts.AddChild();
        _chExperts.person_id = auditorTE.id;
        _chExperts.person_fullname = auditorTE.fullname;
        _chExperts.person_position_name =
auditorTE.position_name;
        _chExperts.role = auditorTE.position_name;
        _chExperts.responsible = 0;

        new_AA_ID = new_AA.DocID;
        new_AA_TE.id = new_AA_ID;

        new_AA.Save();

tools_ass.generate_assessment_plan(OptInt(new_AA_TE.id), false,
false, false, undefined, undefined, null);

        if (OptInt(new_AA_TE.id) == undefined) {
            oRes.error = 1;
            oRes.errorText = "Ошибка - процедура оценки не
создана.";
            return oRes
        } else {
            new_AA_TE.AssignElem(etalon_TE);
            oRes.result = new_AA.DocID;
            return oRes;
        }
    } catch (error) {
        oRes.error = 1;
        oRes.errorText = error;
        return oRes
    }
}

function createAssessmentAppraiseByCode(etalon_aap_code,
auditor_user_id, evaluate_user_id) {
    oRes = new Object();
    oRes.error = 0;
    oRes.errorText = "";
    oRes.result = undefined;

    try {

```

```

        _findAAID = ArrayOptFirstElem(XQuery("for $elem in
assessment_appraises where $elem/code = '" + etalon_aap_code + "'
and $elem/is_model=true() return $elem"));

        if (_findAAID != undefined) {
            _createAA =
createAssessmentAppraiseById(_findAAID.id, auditor_user_id,
evaluate_user_id);
            if (_createAA.error != 0) {
                oRes.error = 1;
                oRes.errorText = _createAA.errorText;
                return oRes
            } else {
                oRes.result = _createAA.result;
                return oRes;
            }
        } else {
            oRes.error = 1;
            oRes.errorText = 'Не найдена эталонная процедура
оценки с кодом ' + etalon_aap_code;
            return oRes
        }

    } catch (error) {
        oRes.error = 1;
        oRes.errorText = error;
        return oRes
    }
}

function createAssessmentAppraiseGroupById(etalon_aap_id,
auditor_user_id, request_id) {
    oRes = new Object();
    oRes.error = 0;
    oRes.errorText = "";
    oRes.result = undefined;

    checkIncomingData = '';
    if (ArrayOptFirstElem(XQuery("for $elem in
assessment_appraises where $elem/id = " + OptInt(etalon_aap_id) +
" return $elem"))) == undefined) {
        checkIncomingData += 'Не найдена эталонная процедура
оценки.'
    }
    _auditor = tools.open_doc(OptInt(auditor_user_id));
    if (_auditor != undefined) {
        var auditorTE = _auditor.TopElem;
    } else {
        checkIncomingData += (checkIncomingData != ' ' ? ' ' :
'') + 'Не найден эксперт.';
    }
    _request = tools.open_doc(OptInt(request_id));

```

```

        if (_request != undefined) {
            var requestTE = _request.TopElem;
            if (requestTE.is_group != true) {
                checkIncomingData += 'Заявка не является
групповой.'
            }
        } else {
            checkIncomingData += (checkIncomingData != ' ' ? ' ' :
'') + 'Не найден оцениваемый.';
        }

        if (checkIncomingData != '') {
            oRes.error = 1;
            oRes.errorText = checkIncomingData;
            return oRes
        }

        party = ArrayOptFirstElem(XQuery("sql: SELECT id,
team_name FROM cc_skill_participations WHERE request_id = " +
request_id));

        try {
            new_AA = tools.new_doc_by_name("assessment_appraise",
false);

            new_AA.BindToDb();
            etalon_id = OptInt(etalon_aap_id);
            etalon = OpenDoc(UrlFromDocID(etalon_id));
            etalon_TE = etalon.TopElem
            new_AA_TE = new_AA.TopElem;
            new_AA_TE.AssignElem(etalon_TE);

            new_AA_TE.name += ' ' + party.team_name + " " +
CurDate;

            new_AA_TE.is_model = false;
            new_AA_TE.start_date = Date();
            new_AA_TE.status = '0';
            new_AA_TE.person_id = auditorTE.id;
            new_AA_TE.assessment_object_type = "request"
            new_AA_assessment_object =
new_AA_TE.assessment_objects.AddChild();
            new_AA_assessment_object.assessment_object_id =
request_id;

            _chExperts =
ArrayOptFirstElem(new_AA_TE.participants).customize.experts.AddChi
ld();

            _chExperts.person_id = auditorTE.id;
            _chExperts.person_fullname = auditorTE.fullname;
            _chExperts.person_position_name =
auditorTE.position_name;
            _chExperts.role = auditorTE.position_name;
            _chExperts.responsible = 0;

```

```

        new_AA_ID = new_AA.DocID;
        new_AA_TE.id = new_AA_ID;

        new_AA.Save();

tools_ass.generate_assessment_plan(OptInt(new_AA_TE.id), false,
false, false, undefined, undefined, null);

        if (OptInt(new_AA_TE.id) == undefined) {
            oRes.error = 1;
            oRes.errorText = "Ошибка - процедура оценки не
создана.";
            return oRes
        } else {
            new_AA_TE.AssignElem(etalon_TE);

            new_AssPlan =
tools.new_doc_by_name("assessment_plan", false);
            new_AssPlan.BindToDb();
            new_AssPlan_TE = new_AssPlan.TopElem;

            new_AssPlan_TE.assessment_appraise_id =
new_AA.DocID;

            new_AssPlan_TE.boss_id = auditor_user_id;
            new_AssPlan_TE.department_id = request_id;
            new_AssPlan_TE.department_name = "request";
            new_AssPlan_TE.flag_appraise_department = true;
            new_AssPlan_TE.assessment_object_type = "request";
            new_AssPlan_TE.is_done = false;
            new_AssPlan_TE.flag_is_processed = false;
            new_AssPlan_TE.workflow_id =
new_AA_TE.workflow_id;
            new_AssPlan_TE.workflow_state = 0;
            new_AssPlan_TE.workflow_state_name = 'Заполнение';

            new_AssPlan.Save();

            cp_id =
new_AA_TE.participants[0].assessment_appraise_types[0].competence_
profile_id;

            wf_id = new_AA_TE.workflow_id;
            ap_id = new_AssPlan.DocID;

            teCompProfile =
OpenDoc (UrlFromDocID(cp_id)).TopElem;

            new_PA = tools.new_doc_by_name("pa", false);
            new_PA.BindToDb();
            new_PA_TE = new_PA.TopElem;

            new_PA_TE.flag_appraise_department = true;
            new_PA_TE.assessment_object_type = "request";
            new_PA_TE.department_id = request_id;

```

```

        new_PA_TE.department_name = "request";
        new_PA_TE.assessment_plan_id = ap_id;
        new_PA_TE.assessment_appraise_id = new_AA.DocID;
        new_PA_TE.expert_person_id = auditor_user_id;
        new_PA_TE.status = 'expert';
        new_PA_TE.assessment_appraise_type =
'competence_appraisal';
        new_PA_TE.competence_profile_id = cp_id;
        new_PA_TE.workflow_id = wf_id;
        new_PA_TE.workflow_state = 0;
        new_PA_TE.workflow_state_name = 'Заполнение';

new_PA_TE.competences.AssignElem(teCompProfile.competences);
        new_PA.Save();

        oRes.result = new_AA.DocID;
        return oRes;
    }
} catch (error) {
    oRes.error = 1;
    oRes.errorText = error.message;
    return oRes
}
}

function createAssessmentAppraiseGroupByCode(etalon_aap_code,
auditor_user_id, request_id) {
    oRes = new Object();
    oRes.error = 0;
    oRes.errorText = "";
    oRes.result = undefined;

    try {

        _findAAID = ArrayOptFirstElem(XQuery("for $elem in
assessment_appraises where $elem/code = '" + etalon_aap_code + "'
and $elem/is_model=true() return $elem"));

        if (_findAAID != undefined) {
            _createAA =
createAssessmentAppraiseGroupById(_findAAID.id, auditor_user_id,
request_id);

            if (_createAA.error != 0) {
                oRes.error = 1;
                oRes.errorText = _createAA.errorText;
                return oRes
            } else {
                oRes.result = _createAA.result;
                return oRes;
            }

        } else {

```

```

        oRes.error = 1;
        oRes.errorText = 'Не найдена эталонная процедура
оценки с кодом ' + etalon_aap_code;
        return oRes
    }

    } catch (error) {
        oRes.error = 1;
        oRes.errorText = error;
        return oRes
    }
}

function createReport(nominationCode) {
    oRes = new Object();
    oRes.error = 0;
    oRes.errorText = "";
    oRes.result = [];

    try {
        sqlReq = " declare @codeStage varchar(50) = '" +
nominationCode + "'; " +
        " SELECT " +
        "
aa.data.value('(/assessment_appraise/participants/participant/asse
ssment_appraise_types/assessment_appraise_type/competence_profile_
id)[1]', 'NVARCHAR(100)') AS competence_profile_id " +
        " FROM assessment_appraises aas " +
        " INNER JOIN assessment_appraise aa ON aa.id =
aas.id " +
        " WHERE aas.code = @codeStage " +
        " AND aas.is_model = 1 ";

        compProfileId = ArrayOptFirstElem(XQuery("sql: " +
sqlReq), {
            competence_profile_id: 0,
        }).competence_profile_id;

        var compProf =
OpenDoc (UrlFromDocID (OptInt (compProfileId)));
        if (compProf == undefined) {
            oRes.error = 0;
            oRes.errorText = "Не удалось открыть профиль
компетенций с ид " + string(compProfileId);
            return oRes;
        }

        var competences = compProf.TopElem.competences;
        for (comp in competences) {
            compDoc =
OpenDoc (UrlFromDocID (OptInt (comp.competence_id)));
            scalesStr = "";
            scaleElems = compDoc.TopElem.scales;

```

```

        for (scale in scaleElems) {
            scalesStr += String(scale.name) + " - " +
String(scale.desc) + "\n";
        }
        indicatorElem = comp.indicators;
        for (indicator in indicatorElem) {
            tempObj = new Object();
            tempObj = {
                nomination_id: String(compDoc.TopElem.id),
                nomination_name:
String(compDoc.TopElem.name),
                indicator_id:
String(indicator.indicator_id),
                indicator_name:
String(indicator.indicator_id.OptForeignElem.name),
                scale: scalesStr,
            };
            oRes.result.push(tempObj);
        }
    }

    return oRes;

} catch (error) {
    oRes.error = 1;
    oRes.errorText = error;
    return oRes
}
}

function getCompetitionPicts(user_id) {
    var oResult = new Object();

    oResult.error = 0;
    oResult.errorText = "";
    oResult.result = [];

    try {
        if (OptInt(user_id) == undefined) {
            throw "значение параметра user_id, задано некорректно: "
+ user_id;
        }

        var sql =
            "sql: \
            DECLARE @user_id bigint = " +
            user_id +
            "; \
            WITH extracted_data AS ( \
            SELECT \
                rs.id, \
                rs.creation_date, \

```

```

        r.data.value('/resource/file_name')[1]',
'nvarchar(max)') as file_name, \
        r.data.value('/resource/comment')[1]',
'nvarchar(max)') as comment \
        FROM resources rs \
        JOIN resource r ON rs.id = r.id \
        WHERE rs.code = 'skill_competition_pics' \
    ), \
    extracted_likes AS ( \
        SELECT \
            object_id, \
            collaborator_id \
        FROM cc_likes \
        WHERE code = 'skill_competition_like' \
    ), \
    like_info AS ( \
        SELECT \
            object_id, \
            COUNT(*) as like_count, \
            MAX(CASE WHEN collaborator_id = @user_id THEN 1 ELSE 0
END) as user_liked \
        FROM extracted_likes \
        GROUP BY object_id \
    ) \
    SELECT \
        ed.id, \
        ed.creation_date, \
        ed.file_name, \
        ed.comment, \
        ISNULL(li.user_liked, 0) as is_liked, \
        ISNULL(li.like_count, 0) as like_count \
    FROM extracted_data ed \
    LEFT JOIN like_info li ON li.object_id = ed.id \
    ORDER BY ed.creation_date DESC \
    ;";

var sqlResult = XQuery(sql);

oResult.result = sqlResult;

return oResult;
} catch (error) {
    oResult.error = 1;
    oResult.errorText = error + "";

    return oResult;
}
}

function getCompetitionPreviews(user_id) {
    var oResult = new Object();

    oResult.error = 0;

```

```

oResult.errorText = "";
oResult.result = [];

try {
    if (OptInt(user_id) == undefined) {
        throw "значение параметра user_id, задано некорректно: "
+ user_id;
    }

    var sql =
        "sql: \
        DECLARE @user_id bigint = " +
        user_id +
        "; \
        WITH extracted_data AS ( \
            SELECT \
                rs.id, \
                r.data.value('/resource/comment)[1]',
'nvarchar(max)') as comment \
            FROM resources rs \
            JOIN resource r ON rs.id = r.id \
            WHERE rs.code = 'skill_competition_videos_preview'
\
            AND r.data.value('/resource/comment)[1]',
'nvarchar(max)') IS NOT NULL \
            AND
LTRIM(RTRIM(r.data.value('/resource/comment)[1]',
'nvarchar(max)')) <> '' \
            AND TRY_CONVERT(bigint,
LTRIM(RTRIM(r.data.value('/resource/comment)[1]',
'nvarchar(max)')))) IS NOT NULL \
        ), \
        extracted_likes AS ( \
            SELECT \
                object_id, \
                collaborator_id \
            FROM cc_likes \
            WHERE code = 'skill_competition_like' \
        ), \
        like_info AS ( \
            SELECT \
                object_id, \
                COUNT(*) as like_count, \
                MAX(CASE WHEN collaborator_id = @user_id THEN 1
ELSE 0 END) as user_liked \
            FROM extracted_likes \
            GROUP BY object_id \
        ) \
        SELECT \
            ed.id, \
            r.id as video_id, \
            r.created as creation_date, \

```

```

        r.data.value('/resource/file_name')[1]',
'nvarchar(max)') as file_name, \
        r.data.value('/resource/comment')[1]',
'nvarchar(max)') as comment, \
        ISNULL(li.user_liked, 0) as is_liked, \
        ISNULL(li.like_count, 0) as like_count \
        FROM extracted_data ed \
        JOIN resource r ON ed.comment = r.id \
        LEFT JOIN like_info li ON li.object_id = r.id \
        ORDER BY r.created DESC \
";

var sqlResult = XQuery(sql);

oResult.result = sqlResult;

return oResult;
} catch (error) {
oResult.error = 1;
oResult.errorText = error + "";

return oResult;
}
}

function setLike(user_id, object_id) {
var oResult = new Object();

oResult.error = 0;
oResult.errorText = "";
oResult.result = null;

try {
    if (OptInt(user_id) == undefined) {
        throw "значение параметра user_id, задано некорректно: "
+ user_id;
    }

    if (OptInt(object_id) == undefined) {
        throw "значение параметра object_id, задано некорректно: "
+ object_id;
    }

    var sqlCheckObject =
        "sql: \
        DECLARE @object_id bigint = " +
        object_id +
        "; \
        SELECT \
            id \
        FROM resources \
        WHERE id = @object_id \
        ";
};

```

```

        var sqlCheckObjectResult =
ArrayOptFirstElem(XQuery(sqlCheckObject));

        if (sqlCheckObjectResult == undefined) {
            throw "Не удалось найти объект с id - " + object_id;
        }

        var sqlCheckLike =
            "sql: \
            DECLARE @user_id bigint = " +
            user_id +
            "; \
            DECLARE @object_id bigint = " +
            object_id +
            "; \
            SELECT \
                id \
            FROM cc_likes \
            WHERE collaborator_id = @user_id AND object_id =
@object_id \
            ";

        var sqlCheckLikeResult =
ArrayOptFirstElem(XQuery(sqlCheckLike));

        if (sqlCheckLikeResult != undefined) {
            throw (
                "Лайк на объект с id - " +
                object_id +
                " пользователем с id - " +
                user_id +
                " уже создан"
            );
        }

        var newLikeDoc = tools.new_doc_by_name("cc_like", false);

        newLikeDoc.TopElem.code = "skill_competition_like";
        newLikeDoc.TopElem.name =
            "Лайк объекту изображению или видео на странице Конкурс
Лиги Роста";
        newLikeDoc.TopElem.collaborator_id = user_id;
        newLikeDoc.TopElem.object_id = object_id;
        newLikeDoc.TopElem.date = Date();

        newLikeDoc.BindToDb();
        newLikeDoc.Save();

        oResult.result = newLikeDoc.DocID + "";

        return oResult;
    } catch (error) {

```

```

        oResult.error = 1;
        oResult.errorText = error + "";

        return oResult;
    }
}

function removeLike(user_id, object_id) {
    var oResult = new Object();

    oResult.error = 0;
    oResult.errorText = "";
    oResult.result = null;

    try {
        if (OptInt(user_id) == undefined) {
            throw "значение параметра user_id, задано некорректно: "
+ user_id;
        }

        if (OptInt(object_id) == undefined) {
            throw "значение параметра object_id, задано некорректно:
" + object_id;
        }

        var sqlCheckLike =
            "sql: \
            DECLARE @user_id bigint = " +
            user_id +
            "; \
            DECLARE @object_id bigint = " +
            object_id +
            "; \
            SELECT \
                id \
            FROM cc_likes \
            WHERE collaborator_id = @user_id AND object_id =
@object_id \
            ";

        var sqlCheckLikeResult =
            ArrayOptFirstElem(XQuery(sqlCheckLike));

        if (sqlCheckLikeResult == undefined) {
            throw (
                "Лайк на объект с id - " +
                object_id +
                " помтавленный пользователем с id - " +
                user_id +
                " не найден"
            );
        }
    }
}

```

```

DeleteDoc (UrlFromDocID (OptInt (sqlCheckLikeResult.id)));

oResult.result = sqlCheckLikeResult.id + "";

return oResult;
} catch (error) {
oResult.error = 1;
oResult.errorText = error + "";

return oResult;
}
}

```

					ПГУ 1.090501.12.001 ПЗ	Лист
						112
Изм.	Лист	№ докум.	Подпись	Дата		

Приложение В

(обязательное)

Модуль проведения многоэтапного отбора автоматизированной системы
управления персоналом
Результаты испытаний

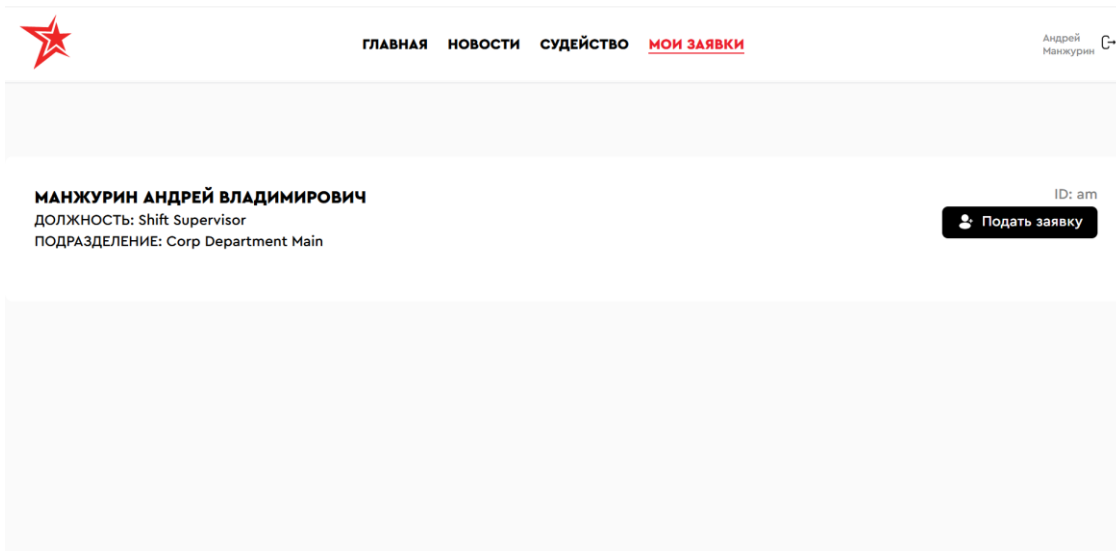


Рисунок В.1 – Страница мои заявки

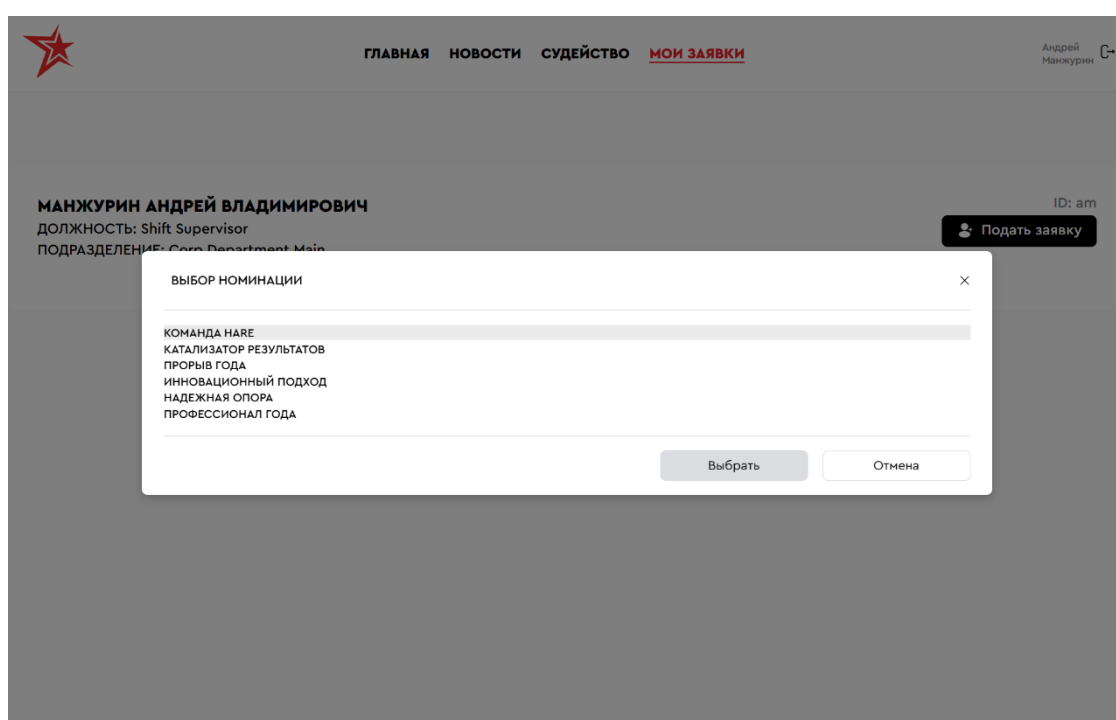


Рисунок В.2 – Выбор номинации для подачи заявки

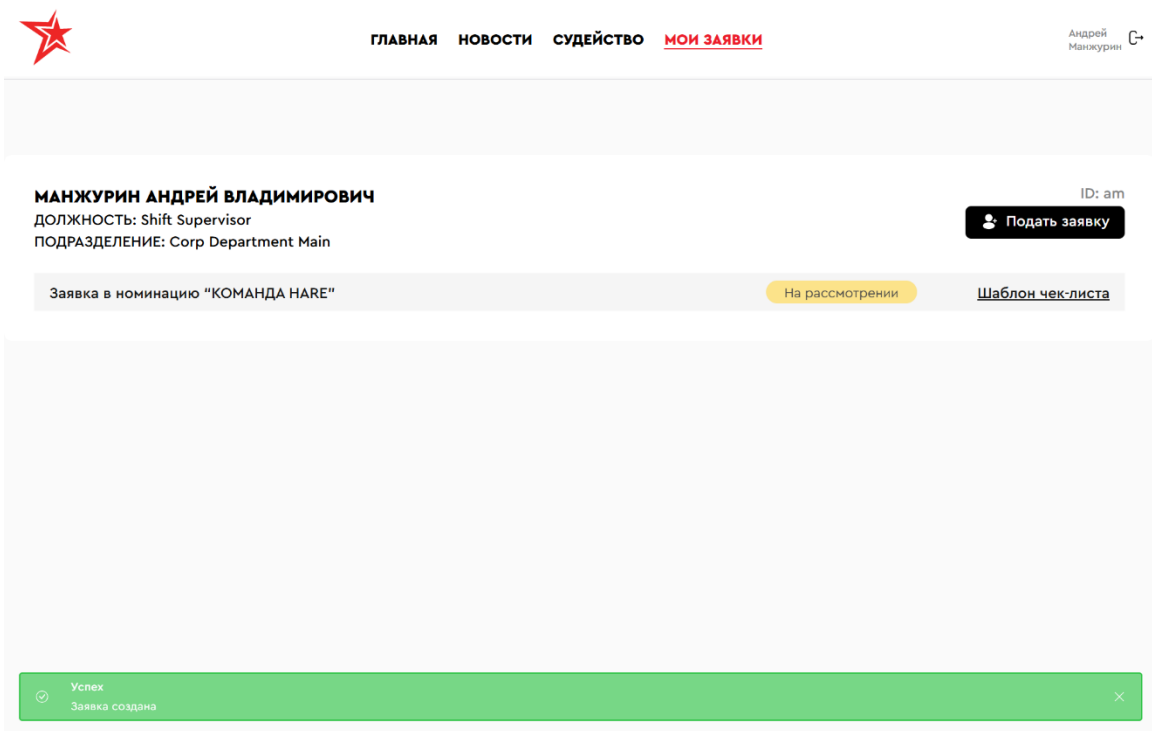


Рисунок В.3 – Заявка по командной номинации успешно подана

<div>🔍 + Новая запись 📄 Фильтр ⚙️ 📄 ...</div>									
Код	Наименование	Заявка	Период Лиги роста	Номинация	Наименование команды	RGM	Area Coach	Market Coach	RSC
	2025 - №662 - Манж	662	2025	КОМАНДА HARE		Лазарев Алексей	Тайх Даниил	Минин Даниил	Зорин Павел Анатол

Рисунок В.4 – Заявка создалась в документе

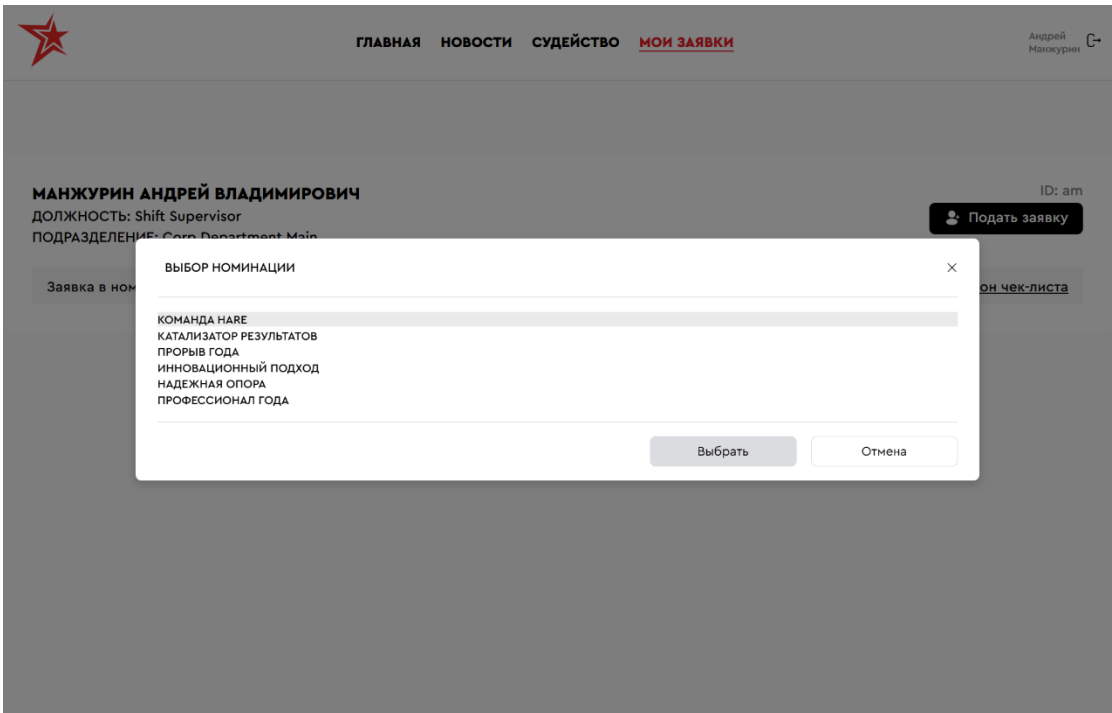


Рисунок В.5 – Повторная подача командной заявки



МАНЖУРИН АНДРЕЙ ВЛАДИМИРОВИЧ

ДОЛЖНОСТЬ: Shift Supervisor

ПОДРАЗДЕЛЕНИЕ: Corp Department Main

ID: am

Подать заявку

Заявка в номинацию "КОМАНДА HARE"

На рассмотрении

[Шаблон чек-листа](#)



Ошибка

Заявка по данной номинации была подана ранее



Рисунок В.6 – Ошибка повторной подачи заявки



МАНЖУРИН АНДРЕЙ ВЛАДИМИРОВИЧ

ДОЛЖНОСТЬ: Shift Supervisor

ПОДРАЗДЕЛЕНИЕ: Corp Department Main

ID: am

Подать заявку

Заявка в номинацию "КОМАНДА HARE"

На рассмотрении

[Шаблон чек-листа](#)

Заявка в номинацию "КАТАЛИЗАТОР РЕЗУЛЬТАТОВ"

На рассмотрении

[Шаблон чек-листа](#)

Заявка в номинацию "НАДЕЖНАЯ ОПОРА"

На рассмотрении

[Шаблон чек-листа](#)



Успех

Заявка создана



Рисунок В.7 – Поданы заявки по номинациям



НА СОГЛАСОВАНИИ

Введите имя сотрудника

КОМАНДА NARE

КАТАЛИЗАТОР РЕЗУЛЬТАТОВ

ПРОРЫВ ГОДА

ИННОВАЦИОННЫЙ ПОДХОД

НАДЕЖНАЯ ОПОРА

ПРОФЕССИОНАЛ ГОДА

Заявка: 2025 – №664 – Манжурин Андрей Владимирович
Номинация: КАТАЛИЗАТОР РЕЗУЛЬТАТОВ

Сотрудник: Манжурин Андрей Владимирович
Должность: Shift Supervisor
Подразделение: Corp Department Main

Принять

Отклонить

1 ЭТАП

2 ЭТАП

3 ЭТАП

Рисунок В.8 – Вкладка согласования индивидуальной заявки



НА СОГЛАСОВАНИИ

Введите имя сотрудника

КОМАНДА NARE

КАТАЛИЗАТОР РЕЗУЛЬТАТОВ

ПРОРЫВ ГОДА

ИННОВАЦИОННЫЙ ПОДХОД

НАДЕЖНАЯ ОПОРА

ПРОФЕССИОНАЛ ГОДА

Заявка: 2025 – №664 – Манжурин Андрей Владимирович
Номинация: КАТАЛИЗАТОР РЕЗУЛЬТАТОВ

Сотрудник: Манжурин Андрей Владимирович
Должность: Shift Supervisor
Подразделение: Corp Department Main

Принять

Отклонить

ПРИНЯТЬ ЗАЯВКУ

×

Вы действительно хотите принять заявку 2025 – №664 – Манжурин Андрей Владимирович ?

Принять

Отмена

1 ЭТАП

2 ЭТАП

Рисунок В.9 – Модальное окно подтверждения заявки



НА СОГЛАСОВАНИИ

Введите имя сотрудника

КОМАНДА NARE 1 КАТАЛИЗАТОР РЕЗУЛЬТАТОВ 0 ПРОРЫВ ГОДА 0 ИННОВАЦИОННЫЙ ПОДХОД 0 НАДЕЖНАЯ ОПОРА 1 ПРОФЕССИОНАЛ ГОДА 0

Список заявок пуст

1 ЭТАП

Введите имя сотрудника

Успех 0 1 0 0 0 0
КОМАНДА NARE принята КАТАЛИЗАТОР РЕЗУЛЬТАТОВ ПРОРЫВ ГОДА ИННОВАЦИОННЫЙ ПОДХОД НАДЕЖНАЯ ОПОРА ПРОФЕССИОНАЛ ГОДА

Рисунок В.10 – Индивидуальная заявка принята



НА СОГЛАСОВАНИИ

Введите имя сотрудника

КОМАНДА NARE 1 КАТАЛИЗАТОР РЕЗУЛЬТАТОВ 0 ПРОРЫВ ГОДА 0 ИННОВАЦИОННЫЙ ПОДХОД 0 НАДЕЖНАЯ ОПОРА 1 ПРОФЕССИОНАЛ ГОДА 0

Список заявок пуст

1 ЭТАП

Введите имя сотрудника

0 1 0 0 0 0
КОМАНДА NARE КАТАЛИЗАТОР РЕЗУЛЬТАТОВ ПРОРЫВ ГОДА ИННОВАЦИОННЫЙ ПОДХОД НАДЕЖНАЯ ОПОРА ПРОФЕССИОНАЛ ГОДА

Манжури́н Андрей Влади́мирович
Corp Department Main

Рисунок В.11 – Заявка перешла на следующий этап

Изм.	Лист	№ докум.	Подпись	Дата

ПГУ 1.090501.12.001 ПЗ

Лист

118



НА СОГЛАСОВАНИИ

Введите имя сотрудника

КОМАНДА NARE

КАТАЛИЗАТОР РЕЗУЛЬТАТОВ

ПРОРЫВ ГОДА

ИННОВАЦИОННЫЙ ПОДХОД

НАДЕЖНАЯ ОПОРА

ПРОФЕССИОНАЛ ГОДА

Заявка: 2025 – №665 – Манжури́н Андрей Владимирович

Номинация: НАДЕЖНАЯ ОПОРА

Сотрудник: Манжури́н Андрей Владимирович
Должность: Shift Supervisor
Подразделение: Corp Department Main

Принять

Отклонить

1 ЭТАП

Введите имя сотрудника

Рисунок В.12 – Индивидуальная заявка на вкладке согласования



НА СОГЛАСОВАНИИ

Введите имя сотрудника

КОМАНДА NARE

КАТАЛИЗАТОР РЕЗУЛЬТАТОВ

ПРОРЫВ ГОДА

ИННОВАЦИОННЫЙ ПОДХОД

НАДЕЖНАЯ ОПОРА

ПРОФЕССИОНАЛ ГОДА

Заявка: 2025 – №665 – Манжури́н Андрей Владимирович

Номинация: НАДЕЖНАЯ ОПОРА

Сотрудник: Манжури́н Андрей Владимирович
Должность: Shift Supervisor
Подразделение: Corp Department Main

Принять

Отклонить

1 ЭТАП

ОТКЛОНИТЬ ЗАЯВКУ

×

Вы действительно хотите отклонить заявку ?

Отклонить

Отмена

Рисунок В.13 – Модальное окно отклонения заявки



НА СОГЛАСОВАНИИ

Введите имя сотрудника

КОМАНДА НАРЕ

КАТАЛИЗАТОР РЕЗУЛЬТАТОВ

ПРОРЫВ ГОДА

ИННОВАЦИОННЫЙ ПОДХОД

НАДЕЖНАЯ ОПОРА

ПРОФЕССИОНАЛ ГОДА

Заявка: 2025 - №665 - Манжури Андрей Владимирович
Номинация: НАДЕЖНАЯ ОПОРА

Сотрудник: Манжури Андрей Владимирович
Должность: Shift Supervisor
Подразделение: Corp Department Main

Отклонена

Рисунок В.14 – Заявка отклонена

Код	Наименование	Заявка	Период Лиги роста	Номинация	Наименование команды	RGM	Area Coach	Market Coach	RSC
2025 - №662 - Манжури Анд	662	2025	КОМАНДА НАРЕ	Лазарев Алексей	Тайх Даниил	Минин Даниил	Зорин Павел		
2025 - №664 - Манжури Анд	664	2025	КАТАЛИЗАТОР РЕЗУ	Лазарев Алексей	Тайх Даниил	Минин Даниил	Зорин Павел		
2025 - №665 - Манжури Анд	665	2025	НАДЕЖНАЯ ОПОРА	Лазарев Алексей	Тайх Даниил	Минин Даниил	Зорин Павел		
2025 - №668 - Теслов Михаил	668	2025	КОМАНДА НАРЕ	Лазарев Алексей	Тайх Даниил	Минин Даниил	Зорин Павел		
2025 - №669 - Теслов Михаил	669	2025	КАТАЛИЗАТОР РЕЗУ	Лазарев Алексей	Тайх Даниил	Минин Даниил	Зорин Павел		
2025 - №670 - Теслов Михаил	670	2025	ПРОРЫВ ГОДА	Лазарев Алексей	Тайх Даниил	Минин Даниил	Зорин Павел		
2025 - №671 - Теслов Михаил	671	2025	НАДЕЖНАЯ ОПОРА	Лазарев Алексей	Тайх Даниил	Минин Даниил	Зорин Павел		
2025 - №672 - Теслов Михаил	672	2025	ПРОФЕССИОНАЛ ГО	Лазарев Алексей	Тайх Даниил	Минин Даниил	Зорин Павел		
2025 - №673 - Зимина Анаста	673	2025	КОМАНДА НАРЕ	Лазарев Алексей	Тайх Даниил	Минин Даниил	Зорин Павел		
2025 - №674 - Зимина Анаста	674	2025	КАТАЛИЗАТОР РЕЗУ	Лазарев Алексей	Тайх Даниил	Минин Даниил	Зорин Павел		
2025 - №675 - Зимина Анаста	675	2025	НАДЕЖНАЯ ОПОРА	Лазарев Алексей	Тайх Даниил	Минин Даниил	Зорин Павел		
2025 - №676 - Зимина Анаста	676	2025	ПРОФЕССИОНАЛ ГО	Лазарев Алексей	Тайх Даниил	Минин Даниил	Зорин Павел		
2025 - №677 - Лисицкий Вад	677	2025	КОМАНДА НАРЕ	Лазарев Алексей	Тайх Даниил	Минин Даниил	Зорин Павел		
2025 - №678 - Лисицкий Вад	678	2025	ПРОРЫВ ГОДА	Лазарев Алексей	Тайх Даниил	Минин Даниил	Зорин Павел		
2025 - №679 - Лисицкий Вад	679	2025	ПРОФЕССИОНАЛ ГО	Лазарев Алексей	Тайх Даниил	Минин Даниил	Зорин Павел		
2025 - №680 - Лисицкий Вад	680	2025	ИННОВАЦИОННЫЙ	Лазарев Алексей	Тайх Даниил	Минин Даниил	Зорин Павел		
2025 - №681 - Вдовина Алине	681	2025	КОМАНДА НАРЕ	Лазарев Алексей	Тайх Даниил	Минин Даниил	Зорин Павел		
2025 - №682 - Вдовина Алине	682	2025	ПРОФЕССИОНАЛ ГО	Лазарев Алексей	Тайх Даниил	Минин Даниил	Зорин Павел		
2025 - №683 - Вдовина Алине	683	2025	ИННОВАЦИОННЫЙ	Лазарев Алексей	Тайх Даниил	Минин Даниил	Зорин Павел		
2025 - №684 - Закурдаев Кири	684	2025	КОМАНДА НАРЕ	Лазарев Алексей	Тайх Даниил	Минин Даниил	Зорин Павел		
2025 - №685 - Закурдаев Кири	685	2025	ПРОРЫВ ГОДА	Лазарев Алексей	Тайх Даниил	Минин Даниил	Зорин Павел		
2025 - №686 - Захаров Дмитр	686	2025	КОМАНДА НАРЕ	Лазарев Алексей	Тайх Даниил	Минин Даниил	Зорин Павел		
2025 - №687 - Захаров Дмитр	687	2025	КАТАЛИЗАТОР РЕЗУ	Лазарев Алексей	Тайх Даниил	Минин Даниил	Зорин Павел		
2025 - №688 - Захаров Дмитр	688	2025	НАДЕЖНАЯ ОПОРА	Лазарев Алексей	Тайх Даниил	Минин Даниил	Зорин Павел		
2025 - №689 - Соколов Алекс	689	2025	КОМАНДА НАРЕ	Лазарев Алексей	Тайх Даниил	Минин Даниил	Зорин Павел		
2025 - №690 - Соколов Алекс	690	2025	КАТАЛИЗАТОР РЕЗУ	Лазарев Алексей	Тайх Даниил	Минин Даниил	Зорин Павел		

Рисунок В.15 – Создано множество заявок

НА СОГЛАСОВАНИИ

Введите имя сотрудника

Управлять командой

7

3

3

2

4

4

КОМАНДА HARE

КАТАЛИЗАТОР РЕЗУЛЬТАТОВ

ПРОРЫВ ГОДА

ИННОВАЦИОННЫЙ ПОДХОД

НАДЕЖНАЯ ОПОРА

ПРОФЕССИОНАЛ ГОДА

Заявка: 2025 – №662 – Манжури́н Андрей Владимирович

Номинация: КОМАНДА HARE

Сотрудник: Манжури́н Андрей Владимирович

Должность: Shift Supervisor

Подразделение: Corp Department Main

На рассмотрении

Заявка: 2025 – №668 – Теслов Михаил Владимирович

Номинация: КОМАНДА HARE

Сотрудник: Теслов Михаил Владимирович

Должность: Dispatcher

Подразделение: Corp Department Main

На рассмотрении

Заявка: 2025 – №673 – Зими́на Анастасия Александровна

Номинация: КОМАНДА HARE

Сотрудник: Зими́на Анастасия Александровна

Должность: Accountant

Подразделение: Corp Department Main

На рассмотрении

Рисунок В.16 – Командные заявки все одобрены

ГЛАВНАЯ

НОВОСТИ

СУДЕЙСТВО

МОИ ЗАЯВКИ

Алексей Лазарев

Назад

ТЕКУЩИЕ КОМАНДЫ

Создать команду

Рисунок В.17 – Страница управления командой

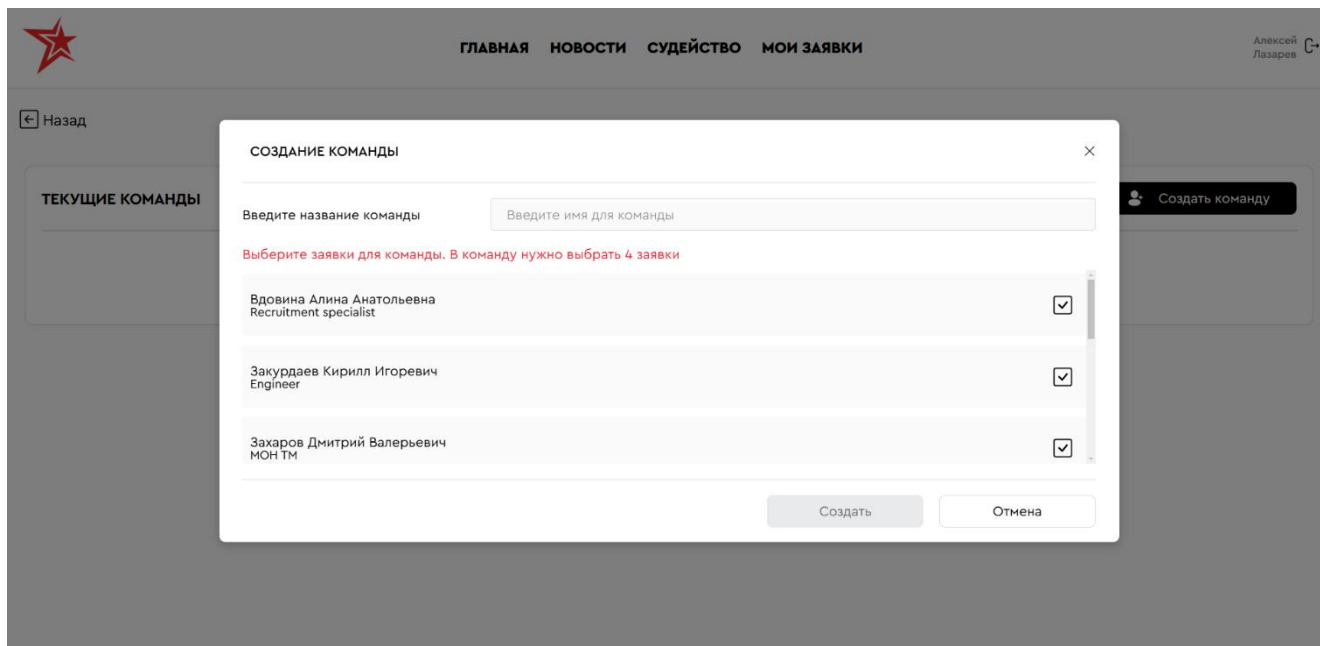


Рисунок В.18 – Модальное окно формирования команды

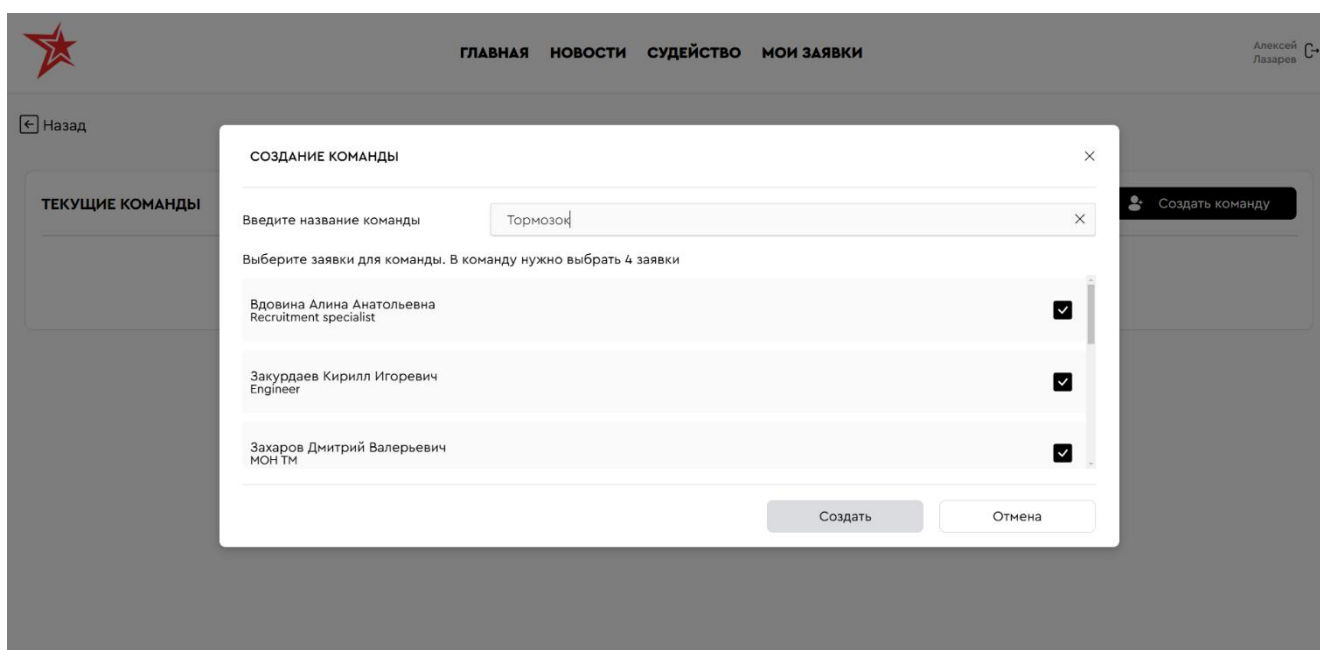


Рисунок В.19 – Заполнение полей формирования команды



Назад

ТЕКУЩИЕ КОМАНДЫ

Создать команду

Команда: Тормозок
Номинация: КОМАНДА HARE



Сотрудник: Вдовина Алина Анатольевна
Должность: Recruitment specialist
Подразделение: Corp Department Main

Сотрудник: Закурдаев Кирилл Игоревич
Должность: Engineer
Подразделение: Corp Department Main

Сотрудник: Захаров Дмитрий Валерьевич
Должность: МОН ТМ
Подразделение: Corp Department Main

Сотрудник: Лисицкий Вадим Сергеевич

Рисунок В.20 – Команда создана



Назад

ТЕКУЩИЕ КОМАНДЫ

Создать команду

Команда: Тормозок
Номинация: КОМАНДА HARE

Сотрудник: Вдовина Алина Анатольевна
Должность: Recruitment specialist
Подразделение: Corp Department Main

Сотрудник: Закурдаев Кирилл Игоревич
Должность: Engineer
Подразделение: Corp Department Main

Сотрудник: Захаров Дмитрий Валерьевич
Должность: МОН ТМ
Подразделение: Corp Department Main

Сотрудник: Лисицкий Вадим Сергеевич

РЕДАКТИРОВАНИЕ КОМАНДЫ

Введите название команды

Новая линия

Выберите заявки для команды. В команду нужно выбрать 4 заявки

Закурдаев Кирилл Игоревич
Engineer



Захаров Дмитрий Валерьевич
МОН ТМ



Лисицкий Вадим Сергеевич
Technologist



Зими́на Анастасия Александровна
Accountant



Сохранить

Отмена

Рисунок В.21 – Модальное окно редактирования команды

Изм.	Лист	№ докум.	Подпись	Дата

ПГУ 1.090501.12.001 ПЗ

Лист

123



Назад

ТЕКУЩИЕ КОМАНДЫ

Создать команду

Команда: Новая линия
Номинация: КОМАНДА HARE

Сотрудник: Вдовина Алина Анатольевна
Должность: Recruitment specialist
Подразделение: Corp Department Main

Сотрудник: Закурдаев Кирилл Игоревич
Должность: Engineer
Подразделение: Corp Department Main

Сотрудник: Захаров Дмитрий Валерьевич
Должность: МОН ТМ
Подразделение: Corp Department Main

Сотрудник: Зими́на Анастасия Александровна

Успех
Заявка отредактирована

Рисунок В.22 – Команда отредактирована

НА СОГЛАСОВАНИИ

Введите имя сотрудника

Управлять командой

КОМАНДА HAREКАТАЛИЗАТОР РЕЗУЛЬТАТОВПРОРЫВ ГОДАИННОВАЦИОННЫЙ ПОДХОДНАДЕЖНАЯ ОПОРАПРОФЕССИОНАЛ ГОДА

Заявка: 2025 – №677 – Лисицкий Вадим Сергеевич
Номинация: КОМАНДА HARE
Сотрудник: Лисицкий Вадим Сергеевич
Должность: Technologist
Подразделение: Corp Department Main

На рассмотрении

Заявка: 2025 – №681 – Вдовина Алина Анатольевна
Номинация: КОМАНДА HARE
Сотрудник: Вдовина Алина Анатольевна
Должность: Recruitment specialist
Подразделение: Corp Department Main
Команда: Новая линия

В команде

Заявка: 2025 – №684 – Закурдаев Кирилл Игоревич
Номинация: КОМАНДА HARE
Сотрудник: Закурдаев Кирилл Игоревич
Должность: Engineer
Подразделение: Corp Department Main
Команда: Новая линия

В команде

Рисунок В.23 – Смена статуса заявок включенных в команду

1 ЭТАП

Введите имя сотрудника

КОМАНДА HAREКАТАЛИЗАТОР РЕЗУЛЬТАТОВПРОРЫВ ГОДАИННОВАЦИОННЫЙ ПОДХОДНАДЕЖНАЯ ОПОРАПРОФЕССИОНАЛ ГОДА

Новая линия
Вдовина Алина Анатольевна, Corp Department Main
Закурдаев Кирилл Игоревич, Corp Department Main
Захаров Дмитрий Валерьевич, Corp Department Main
Зими́на Анастасия Александровна, Corp Department Main

Росток
Лисицкий Вадим Сергеевич, Corp Department Main
Манжурин Андрей Владимирович, Corp Department Main
Соколов Алексей Владимирович, Corp Department Main
Теслов Михаил Владимирович, Corp Department Main

Рисунок В.24 – Командные заявки на 1 этапе вне даты проведения этапа

Дата начала первого этапа

06.05.2025

Дата окончания первого этапа

31.05.2025

Рисунок В.25 – Смена интервалы начала этапа

Подразделение: Corp Department Main

1 ЭТАП [Управлять командой](#)

КОМАНДА НАРЕ КАТАЛИЗАТОР РЕЗУЛЬТАТОВ ПРОРЫВ ГОДА ИННОВАЦИОННЫЙ ПОДХОД НАДЕЖНАЯ ОПОРА ПРОФЕССИОНАЛ ГОДА

Новая линия
Вдовина Алина Анатольевна, Corp Department Main
Закурдаев Кирилл Игоревич, Corp Department Main
Захаров Дмитрий Валерьевич, Corp Department Main
Зинина Анастасия Александровна, Corp Department Main

[Заполнить чек-лист](#)

Росток
Лисицкий Вадим Сергеевич, Corp Department Main
Манжуриин Андрей Владимирович, Corp Department Main
Сokolov Алексей Владимирович, Corp Department Main
Теслов Михаил Владимирович, Corp Department Main

[Заполнить чек-лист](#)

Рисунок В.26 – Появление функциональных кнопок заявок

ГЛАВНАЯ НОВОСТИ **СУДЕЙСТВО** МОИ ЗАЯВКИ

Алексей Лазарев

НА СОГЛАСОВАНИИ

КОМАНДА НАРЕ КАТАЛИЗАТОР РЕЗУЛЬТАТОВ ПРОРЫВ ГОДА ИННОВАЦИОННЫЙ ПОДХОД НАДЕЖНАЯ ОПОРА ПРОФЕССИОНАЛ ГОДА

Заявка: 2025 – №665 – Манжуриин Андрей Владимирович
Номинация: НАДЕЖНАЯ ОПОРА
Сотрудник: Манжуриин Андрей Владимирович
Должность: Shift Supervisor
Подразделение: Corp Department Main

Заявка: 2025 – №688 – Захаров Дмитрий Валерьевич
Номинация: НАДЕЖНАЯ ОПОРА
Сотрудник: Захаров Дмитрий Валерьевич
Должность: МОИ ТМ
Подразделение: Corp Department Main

[Отклонена](#) [Отклонена](#)

ЗАПОЛНИТЬ ЧЕКЛИСТ ЗАЯВКИ

Вы действительно хотите заполнить чеклист для заявки ?

[Заполнить](#) [Отмена](#)

Рисунок В.27 – Модальное окно подтверждения заполнения чек-листа

с 29.05.2025

Заполнение

Оценка завершена

по

Отказ от участия

Оцениваемый сотрудник:
Новая линия

Оценивающий руководитель:
Лазарев Алексей

Оценочные формы

Лазарев Алексей

Ориентация на результат

Вес, %

Оценка

50

-

Оценить

Влияние на других

Вес, %

Оценка

50

-

Оценить

Рисунок В.28 – Страница оценки

Заполнение

Оценка завершена

Отказ от участия

Оцениваемый сотрудник:
Новая линия

Оценивающий руководитель:
Лазарев Алексей

Оценочные формы

✓ Лазарев Алексей

Завершить заполнение

Оценка

83.4

Весовой коэффициент, %

100

Ориентация на результат

Вес, %

Оценка

50

1.67

Оценить

Влияние на других

Вес, %

Оценка

50

2.5

Оценить

Рисунок В.29 – Оценка заполнена

1 ЭТАП

Введите имя сотрудника

Управлять командой

2

4

1

2

2

4

КОМАНДА НАРЕ

КАТАЛИЗАТОР РЕЗУЛЬТАТОВ

ПРОРЫВ ГОДА

ИННОВАЦИОННЫЙ ПОДХОД

НАДЕЖНАЯ ОПОРА

ПРОФЕССИОНАЛ ГОДА

Новая линия

Вдовина Алина Анатольевна, Corp Department Main

Закурдаев Кирилл Игоревич, Corp Department Main

Захаров Дмитрий Валерьевич, Corp Department Main

Зимина Анастасия Александровна, Corp Department Main

Общая оценка: 83.40 %

Заполнить чек-лист

На следующий этап

Отклонить

Алексей Лазарев

29.05.2025

Балл: 83.4 %

Посмотреть чек-лист

Росток

Лисицкий Вадим Сергеевич, Corp Department Main

Манжурич Андрей Владимирович, Corp Department Main

Сokolov Алексей Владимирович, Corp Department Main

Теслов Михаил Владимирович, Corp Department Main

Заполнить чек-лист

Рисунок В.30 – Отображение баллов заполненного чек-листа

1 ЭТАП

Введите имя сотрудника

Управлять командой

2

4

1

2

2

4

КОМАНДА НАРЕ

КАТАЛИЗАТОР РЕЗУЛЬТАТОВ

ПРОРЫВ ГОДА

ИННОВАЦИОННЫЙ ПОДХОД

НАДЕЖНАЯ ОПОРА

ПРОФЕССИОНАЛ ГОДА

Новая линия

Вдовина Алина Анатольевна, Corp Department Main

Закурдаев Кирилл Игоревич, Corp Department Main

Захаров Дмитрий Валерьевич, Corp Department Main

Зимина Анастасия Александровна, Corp Department Main

Общая оценка: 79.20 %

Заполнить чек-лист

На следующий этап

Отклонить

Алексей Лазарев

29.05.2025

Балл: 83.4 %

Посмотреть чек-лист

Алексей Лазарев

29.05.2025

Балл: 75 %

Посмотреть чек-лист

Росток

Лисицкий Вадим Сергеевич, Corp Department Main

Манжурич Андрей Владимирович, Corp Department Main

Заполнить чек-лист

Рисунок В.31 – Среднее арифметическое 2 оценок

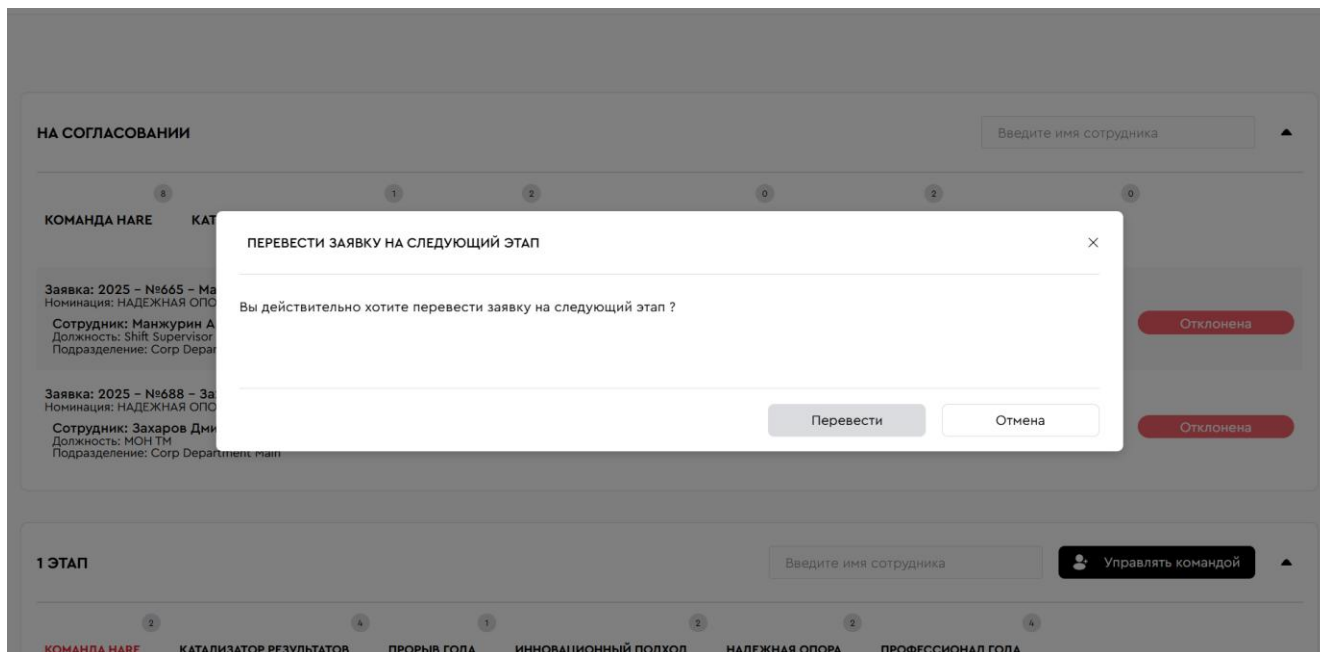


Рисунок В.32 – Модальное окно подтверждения перевода заявки на следующий этап

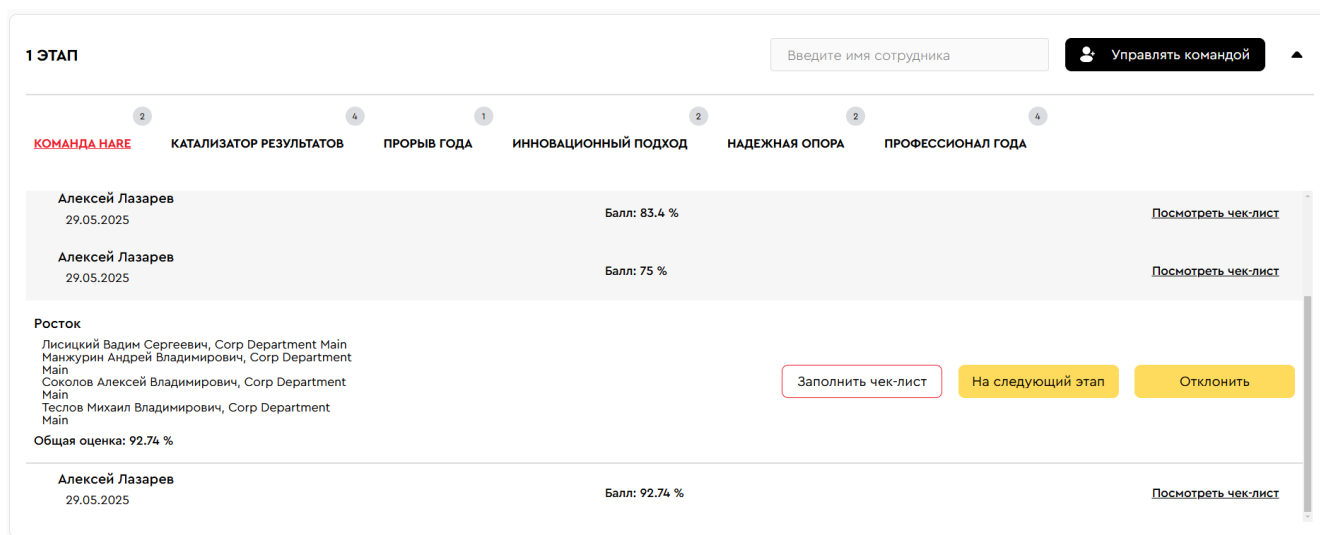


Рисунок В.33 – Отклонение командной заявки «Росток»

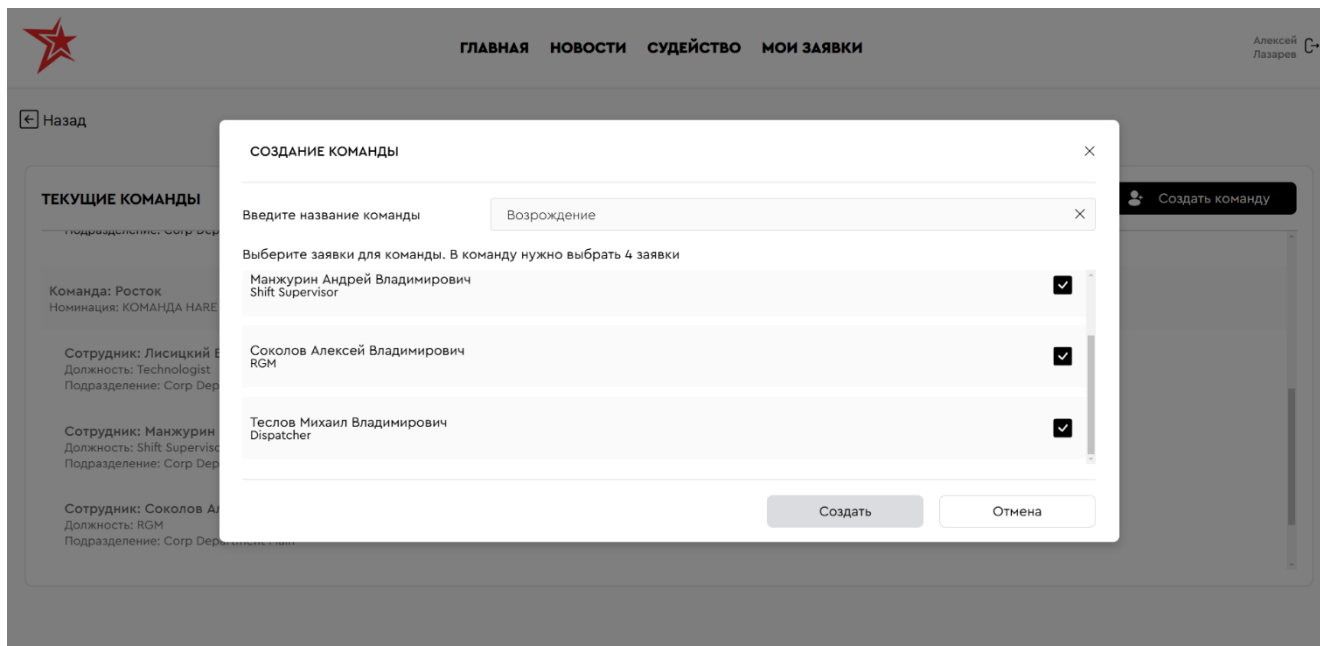


Рисунок В.36 – Выбор наименование и состава команды

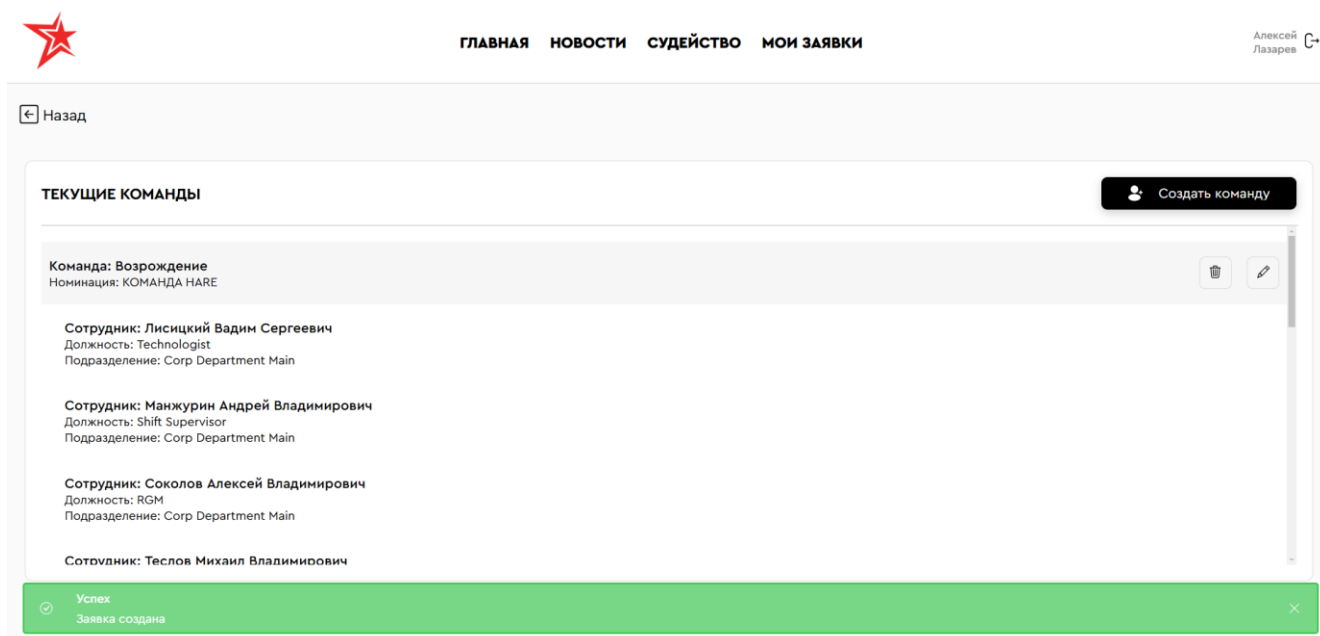


Рисунок В.37 – Команда создана

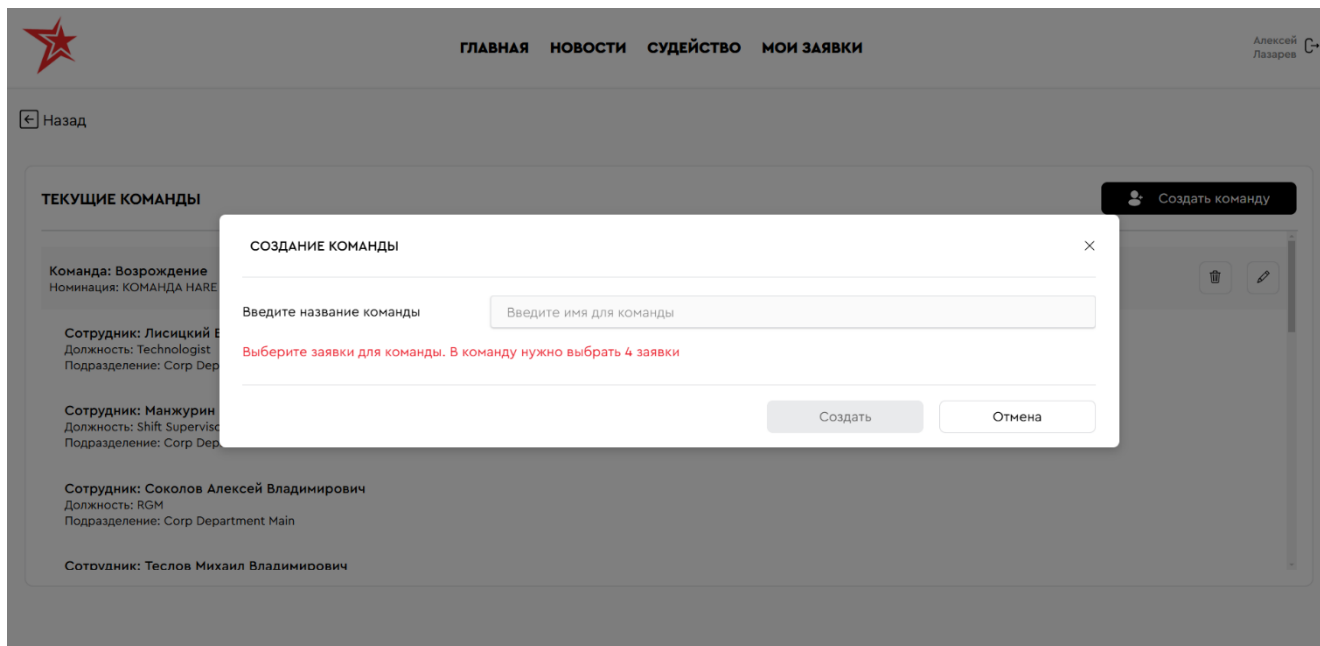


Рисунок В.38 – Включенные заявки в команду пропали из выбора

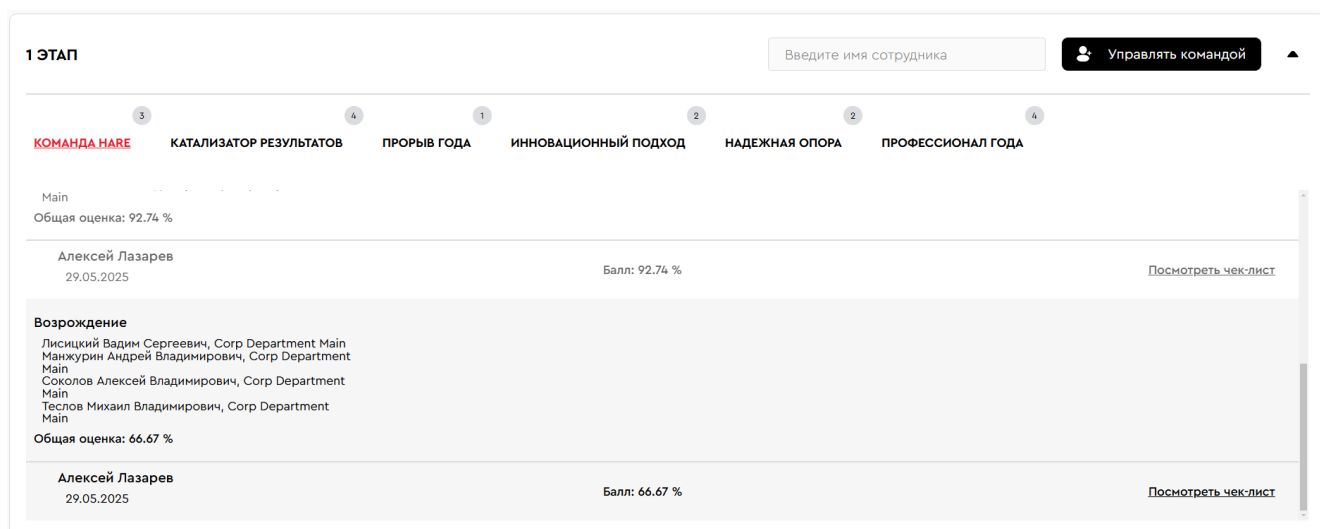


Рисунок В.39 – Команда переведена на 2 этап

2 ЭТАП

Введите имя сотрудника

Управлять командой

200

КОМАНДА НАРЕ

КАТАЛИЗАТОР РЕЗУЛЬТАТОВ

ПРОРЫВ ГОДА

ИННОВАЦИОННЫЙ ПОДХОД

НАДЕЖНАЯ ОПОРА

ПРОФЕССИОНАЛ ГОДА

Новая линия

Вдовина Алина Анатольевна, Corp Department Main
Закурдаев Кирилл Игоревич, Corp Department Main
Захаров Дмитрий Валерьевич, Corp Department Main
Зимина Анастасия Александровна, Corp Department Main

Общая оценка: 79.48 %

Заполнить чек-лист

На следующий этап

Отклонить

Алексей Лазарев
29.05.2025

Балл: 87.5 %

Посмотреть чек-лист

Даниил Тайх
29.05.2025

Балл: 71.46 %

Посмотреть чек-лист

Возрождение

Лисицкий Вадим Сергеевич, Corp Department Main
Манжурич Андрей Владимирович, Corp Department Main

Заполнить чек-лист

Рисунок В.40 – Отображение командных заявок на 2 этапе

2 ЭТАП

Введите имя сотрудника

Управлять командой

200

КОМАНДА НАРЕ

КАТАЛИЗАТОР РЕЗУЛЬТАТОВ

ПРОРЫВ ГОДА

ИННОВАЦИОННЫЙ ПОДХОД

НАДЕЖНАЯ ОПОРА

ПРОФЕССИОНАЛ ГОДА

Новая линия

Вдовина Алина Анатольевна, Corp Department Main
Закурдаев Кирилл Игоревич, Corp Department Main
Захаров Дмитрий Валерьевич, Corp Department Main
Зимина Анастасия Александровна, Corp Department Main

Общая оценка: 79.48 %

Алексей Лазарев
29.05.2025

Балл: 87.5 %

Посмотреть чек-лист

Даниил Тайх
29.05.2025

Балл: 71.46 %

Посмотреть чек-лист

Возрождение

Лисицкий Вадим Сергеевич, Corp Department Main
Манжурич Андрей Владимирович, Corp Department Main

Успех
Заявка принята

3 ЭТАП

Рисунок В.41 – Перевод команды на следующий этап

3 ЭТАП

Введите имя сотрудника

100

КОМАНДА НАРЕ

КАТАЛИЗАТОР РЕЗУЛЬТАТОВ

ПРОРЫВ ГОДА

ИННОВАЦИОННЫЙ ПОДХОД

НАДЕЖНАЯ ОПОРА

ПРОФЕССИОНАЛ ГОДА

Новая линия

Вдовина Алина Анатольевна, Corp Department Main
Закурдаев Кирилл Игоревич, Corp Department Main
Захаров Дмитрий Валерьевич, Corp Department Main
Зимина Анастасия Александровна, Corp Department Main

Рисунок В.42 – Командная заявка на 3 этапе

2 ЭТАП

Введите имя сотрудника

Управлять командой

2

0

0

0

0

0

КОМАНДА НАРЕ

КАТАЛИЗАТОР РЕЗУЛЬТАТОВ

ПРОРЫВ ГОДА

ИННОВАЦИОННЫЙ ПОДХОД

НАДЕЖНАЯ ОПОРА

ПРОФЕССИОНАЛ ГОДА

Алексей Лазарев

29.05.2025

Балл: 87.5 %

Посмотреть чек-лист

Даниил Тайх

29.05.2025

Балл: 71.46 %

Посмотреть чек-лист

Возрождение

Лисицкий Вадим Сергеевич, Corp Department Main

Манжурин Андрей Владимирович, Corp Department Main

Соколов Алексей Владимирович, Corp Department Main

Теслов Михаил Владимирович, Corp Department Main

Общая оценка: 73.04 %

Заполнить чек-лист

На следующий этап

Отклонить

Алексей Лазарев

29.05.2025

Балл: 73.04 %

Посмотреть чек-лист

Рисунок В.43 – Отклонение командной заявки

2 ЭТАП

Введите имя сотрудника

Управлять командой

2

0

0

0

0

0

КОМАНДА НАРЕ

КАТАЛИЗАТОР РЕЗУЛЬТАТОВ

ПРОРЫВ ГОДА

ИННОВАЦИОННЫЙ ПОДХОД

НАДЕЖНАЯ ОПОРА

ПРОФЕССИОНАЛ ГОДА

Алексей Лазарев

29.05.2025

Балл: 87.5 %

Посмотреть чек-лист

Даниил Тайх

29.05.2025

Балл: 71.46 %

Посмотреть чек-лист

Возрождение (команда расформирована)

Лисицкий Вадим Сергеевич, Corp Department Main

Манжурин Андрей Владимирович, Corp Department Main

Соколов Алексей Владимирович, Corp Department Main

Теслов Михаил Владимирович, Corp Department Main

Общая оценка: 73.04 %

Алексей Лазарев

29.05.2025

Балл: 73.04 %

Посмотреть чек-лист

Рисунок В.44 – Команда расформирована

3 ЭТАП

Введите имя сотрудника

Управлять командой

1

0

0

0

0

0

КОМАНДА НАРЕ

КАТАЛИЗАТОР РЕЗУЛЬТАТОВ

ПРОРЫВ ГОДА

ИННОВАЦИОННЫЙ ПОДХОД

НАДЕЖНАЯ ОПОРА

ПРОФЕССИОНАЛ ГОДА

Новая линия

Вдовина Алина Анатольевна, Corp Department Main

Закурдаев Кирилл Игоревич, Corp Department Main

Захаров Дмитрий Валерьевич, Corp Department Main

Зинина Анастасия Александровна, Corp Department Main

Заполнить чек-лист

Рисунок В.45 – Элементы управления заявкой на 3 этапе

НовостиПомощьКарьера

Поиск в библиотеке

Анастасия Зими́на

Мои уведомления

Дата получения	Описание	Перейти
29.05.2025 09:33:20	Поздравляем! В номинации КОМАНДА HARE вы перешли на этап '3 этап'	Открыть
29.05.2025 09:24:36	Поздравляем! В номинации КОМАНДА HARE вы перешли на этап '2 этап'	Открыть
29.05.2025 09:17:56	Поздравляем! В номинации ПРОФЕССИОНАЛ ГОДА вы перешли на '1 этап'	Открыть
29.05.2025 09:17:49	Поздравляем! В номинации НАДЕЖНАЯ ОПОРА вы перешли на '1 этап'	Открыть
29.05.2025 09:17:20	Поздравляем! В номинации КАТАЛИЗАТОР РЕЗУЛЬТАТОВ вы перешли на '1 этап'	Открыть
29.05.2025 09:00:35	Поздравляем! В номинации ПРОФЕССИОНАЛ ГОДА вы перешли на 'регистрация'	Открыть
29.05.2025 09:00:33	Поздравляем! В номинации НАДЕЖНАЯ ОПОРА вы перешли на 'регистрация'	Открыть
29.05.2025 09:00:32	Поздравляем! В номинации КАТАЛИЗАТОР РЕЗУЛЬТАТОВ вы перешли на 'регистрация'	Открыть

Рисунок В.46 – Уведомления пользователя

3 ЭТАП

Введите имя сотрудника

Управлять командой

КОМАНДА HAREКАТАЛИЗАТОР РЕЗУЛЬТАТОВПРОРЫВ ГОДАИННОВАЦИОННЫЙ ПОДХОДНАДЕЖНАЯ ОПОРАПРОФЕССИОНАЛ ГОДА

Новая линия

Вдовина Алина Анатольевна, Corp Department Main
Закурдаев Кирилл Игоревич, Corp Department Main
Захаров Дмитрий Валерьевич, Corp Department Main
Зими́на Анастасия Александровна, Corp Department Main

Общая оценка: 90.00 %

Даниил Минин
29.05.2025

Балл: 90 %

Посмотреть чек-лист

Заполнить чек-лист

На следующий этап

Отклонить

Рисунок В.47 – Заполнен чек-лист

НА СОГЛАСОВАНИИ

Введите имя сотрудника

Управлять командой

КОМАНДА HAREКАТАЛИЗАТОР РЕЗУЛЬТАТОВПРОРЫВ ГОДАИННОВАЦИОННЫЙ ПОДХОДНАДЕЖНАЯ ОПОРАПРОФЕССИОНАЛ ГОДА

Заявка: 2025 – №662 – Манжу́рин Андрей Владимирович
Номинация: КОМАНДА HARE

Сотрудник: Манжу́рин Андрей Владимирович
Должность: Shift Supervisor
Подразделение: Corp Department Main

На рассмотрении

Заявка: 2025 – №668 – Теслов Михаил Владимирович
Номинация: КОМАНДА HARE

Сотрудник: Теслов Михаил Владимирович
Должность: Dispatcher
Подразделение: Corp Department Main

На рассмотрении

Заявка: 2025 – №673 – Зими́на Анастасия Александровна
Номинация: КОМАНДА HARE

Сотрудник: Зими́на Анастасия Александровна
Должность: Accountant
Подразделение: Corp Department Main

В команде

Команда: Новая линия

Рисунок В.48 – Заявки не включенные в команду

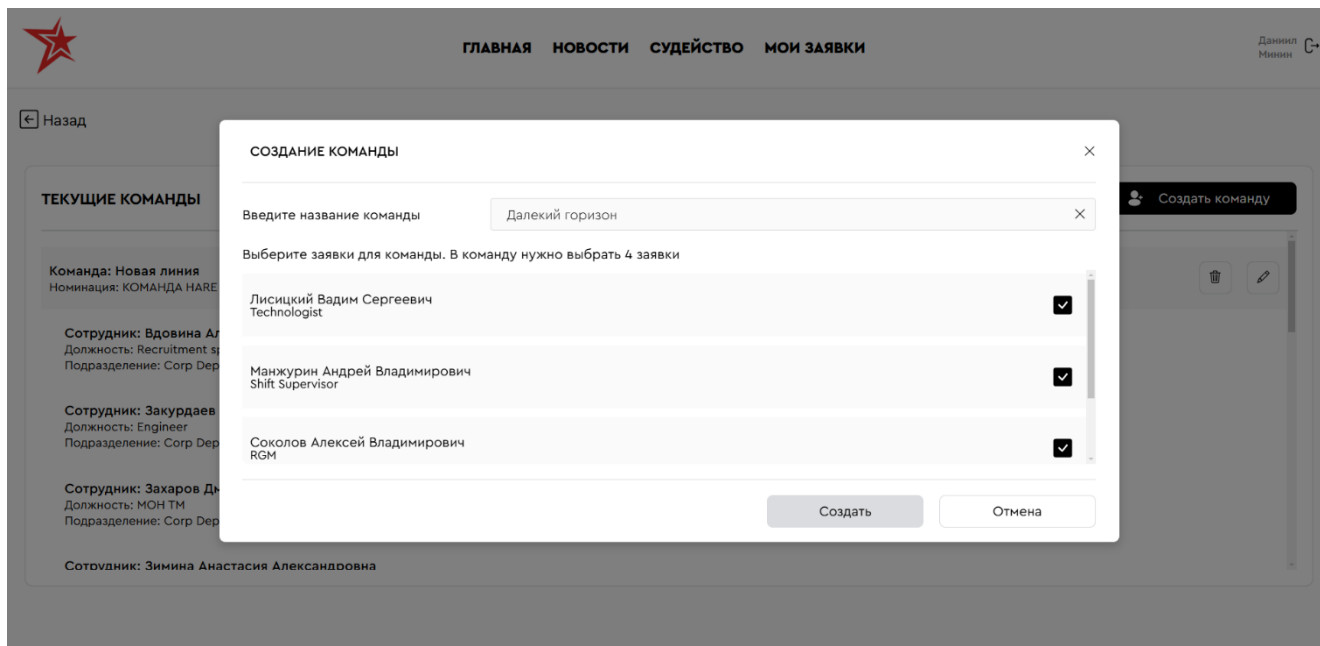


Рисунок В.49 – Формирование команды для 3 этапа

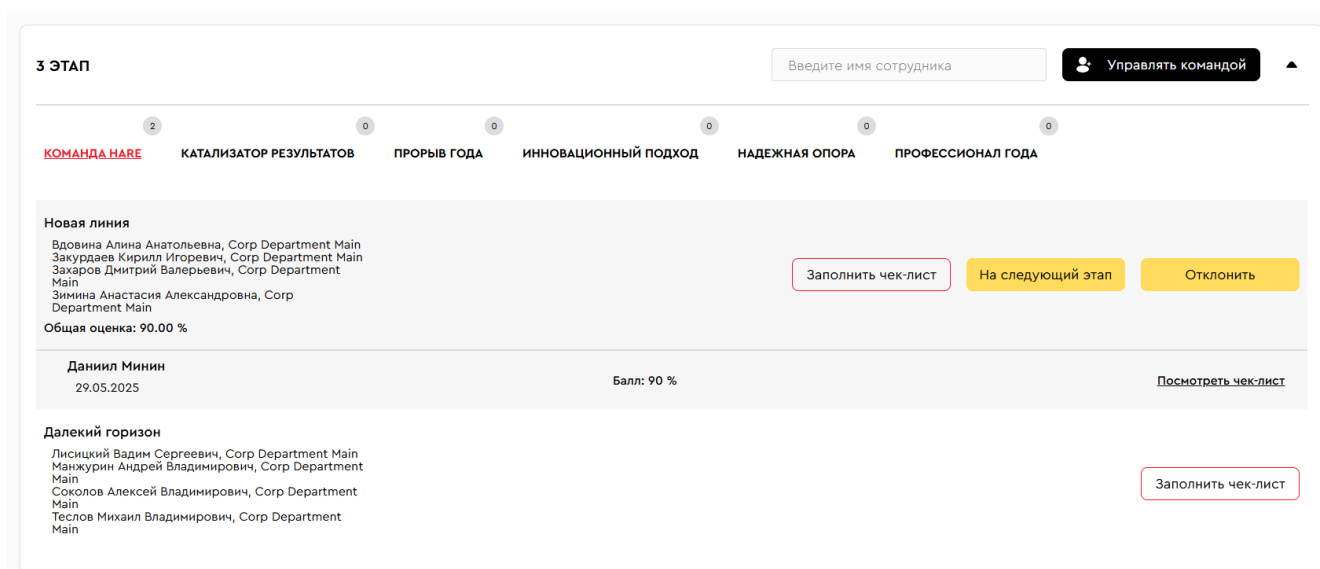


Рисунок В.50 – Команда создана

4 ЭТАП

Введите имя сотрудника

100%

КОМАНДА НАРЕКАТЕЛИ

КАТАЛИЗАТОР РЕЗУЛЬТАТОВ

ПРОРЫВ ГОДА

ИННОВАЦИОННЫЙ ПОДХОД

НАДЕЖНАЯ ОПОРА

ПРОФЕССИОНАЛ ГОДА

Новая линия

Вдовина Алина Анатольевна, Corp Department Main
Закурдаев Кирилл Игоревич, Corp Department Main
Захаров Дмитрий Валерьевич, Corp Department Main
Зимина Анастасия Александровна, Corp Department Main

Заполнить чек-лист

Рисунок В.54 – Возможность заполнить чек-лист на 4 этапе

4 ЭТАП

Введите имя сотрудника

100%

КОМАНДА НАРЕКАТЕЛИ

КАТАЛИЗАТОР РЕЗУЛЬТАТОВ

ПРОРЫВ ГОДА

ИННОВАЦИОННЫЙ ПОДХОД

НАДЕЖНАЯ ОПОРА

ПРОФЕССИОНАЛ ГОДА

Новая линия

Вдовина Алина Анатольевна, Corp Department Main
Закурдаев Кирилл Игоревич, Corp Department Main
Захаров Дмитрий Валерьевич, Corp Department Main
Зимина Анастасия Александровна, Corp Department Main

Общая оценка: 72.22 %

Заполнить чек-лист

Отклонить

Евгений Ларин

29.05.2025

Балл: 80.5 %

Посмотреть чек-лист

Андрей Манжури

29.05.2025

Балл: 52.83 %

Посмотреть чек-лист

Герман Тихомиров

29.05.2025

Балл: 83.33 %

Посмотреть чек-лист

Рисунок В.55 – Заполненные чек-листы специалистами по оценке

4 ЭТАП

Введите имя сотрудника

100%

КОМАНДА НАРЕКАТЕЛИ

КАТАЛИЗАТОР РЕЗУЛЬТАТОВ

ПРОРЫВ ГОДА

ИННОВАЦИОННЫЙ ПОДХОД

НАДЕЖНАЯ ОПОРА

ПРОФЕССИОНАЛ ГОДА

Герман Тихомиров

29.05.2025

Балл: 83.33 %

Посмотреть чек-лист

Далекий горизонт

Лисицкий Вадим Сергеевич, Corp Department Main
Манжури Андрей Владимирович, Corp Department Main
Соколов Алексей Владимирович, Corp Department Main
Теслов Михаил Владимирович, Corp Department Main

Общая оценка: 40.34 %

Заполнить чек-лист

Отклонить

Евгений Ларин

29.05.2025

Балл: 19.5 %

Посмотреть чек-лист

Герман Тихомиров

29.05.2025

Балл: 61.17 %

Посмотреть чек-лист

Рисунок В.56 – Заполнение чек-листов другой команды

4 ЭТАП

Введите имя сотрудника

20000

КОМАНДА НАРЕКАТЕЛИЗАТОР РЕЗУЛЬТАТОВПРОРЫВ ГОДАИННОВАЦИОННЫЙ ПОДХОДНАДЕЖНАЯ ОПОРАПРОФЕССИОНАЛ ГОДА

Андрей Манжури

29.05.2025

Балл: 52.83 %

Посмотреть чек-лист

Герман Тихомиров

29.05.2025

Балл: 83.33 %

Посмотреть чек-лист

Далекий горизонт (команда расформирована)

Лисицкий Вадим Сергеевич, Corp Department Main

Манжури Андрей Владимирович, Corp Department Main

Соколов Алексей Владимирович, Corp Department Main

Теслов Михаил Владимирович, Corp Department Main

Общая оценка: 40.34 %

Евгений Ларин

29.05.2025

Балл: 19.5 %

Посмотреть чек-лист

Герман Тихомиров

29.05.2025

Балл: 61.17 %

Посмотреть чек-лист

Рисунок В.57 – Расформирование команды

4 ЭТАП

Введите имя сотрудника

20000

КОМАНДА НАРЕКАТЕЛИЗАТОР РЕЗУЛЬТАТОВПРОРЫВ ГОДАИННОВАЦИОННЫЙ ПОДХОДНАДЕЖНАЯ ОПОРАПРОФЕССИОНАЛ ГОДА

Новая линия

Вдовина Алина Анатольевна, Corp Department Main

Закурдаев Кирилл Игоревич, Corp Department Main

Захаров Дмитрий Валерьевич, Corp Department Main

Зимина Анастасия Александровна, Corp Department Main

Общая оценка: 72.22 %

Заполнить чек-лист

На следующий этап

Отклонить

Евгений Ларин

29.05.2025

Балл: 80.5 %

Посмотреть чек-лист

Андрей Манжури

29.05.2025

Балл: 52.83 %

Посмотреть чек-лист

Герман Тихомиров

29.05.2025

Балл: 83.33 %

Посмотреть чек-лист

Далекий горизонт (команда расформирована)

Лисицкий Вадим Сергеевич, Corp Department Main

Манжури Андрей Владимирович, Corp Department Main

Рисунок В.58 – Перевод команды в финал

4 ЭТАП

Введите имя сотрудника

1

0

0

0

0

0

КОМАНДА НАРЕ

КАТАЛИЗАТОР РЕЗУЛЬТАТОВ

ПРОРЫВ ГОДА

ИННОВАЦИОННЫЙ ПОДХОД

НАДЕЖНАЯ ОПОРА

ПРОФЕССИОНАЛ ГОДА

Далекий горизонт (команда расформирована)

Лисицкий Вадим Сергеевич, Corp Department Main

Манжуриин Андрей Владимирович, Corp Department Main

Соколов Алексей Владимирович, Corp Department Main

Теслов Михаил Владимирович, Corp Department Main

Общая оценка: 40.34 %

Евгений Ларин

29.05.2025

Балл: 19.5 %

Посмотреть чек-лист

Герман Тихомиров

29.05.2025

Балл: 61.17 %

Посмотреть чек-лист

ЗАКРЫТА

Введите имя сотрудника

1

0

0

0

0

0

КОМАНДА НАРЕ

КАТАЛИЗАТОР РЕЗУЛЬТАТОВ

ПРОРЫВ ГОДА

ИННОВАЦИОННЫЙ ПОДХОД

НАДЕЖНАЯ ОПОРА

ПРОФЕССИОНАЛ ГОДА

Новая линия

Вдовина Алина Анатольевна, Corp Department Main

Закурдаев Кирилл Игоревич, Corp Department Main

Захаров Дмитрий Валерьевич, Corp Department Main

Зиминая Анастасия Александровна, Corp Department Main

Алексей Лазарев

29.05.2025

Балл: 83.4 %

Алексей Лазарев

29.05.2025

Балл: 75 %

Алексей Лазарев

29.05.2025

Балл: 87.5 %

Даниил Тайх

29.05.2025

Балл: 71.46 %

Даниил Тайх

29.05.2025

Балл: 71.46 %

Рисунок В.59 – Оценка команды завершена

Приложение Г

(обязательное)

Модуль проведения многоэтапного отбора автоматизированной системы
управления персоналом
Презентация

					ПГУ 1.090501.12.001 ПЗ	Лист
						140
Изм.	Лист	№ докум.	Подпись	Дата		

Модуль многоэтапного отбор персонала на платформе Websoft HCM

Автор работы: студент группы 20ВОЭ1
Манжурин А.В.

Научный руководитель: к.т.н., профессор кафедры ВТ
Бикташев Р.А.

ЦЕЛЬ РАБОТЫ

Разработать модуль оценки персонала в системе Websoft HCM, которое позволит автоматизировать подачу заявок, управление командами, реализовать этапность оценки.



АКТУАЛЬНОСТЬ

01. Рост цифровизации HR-процессов
02. Необходимость стандартизированной и объективной оценки кандидатов
03. Отсутствие гибкого модуля оценки в типовой Websoft HCM
04. Значение кадровых решений в крупных организациях



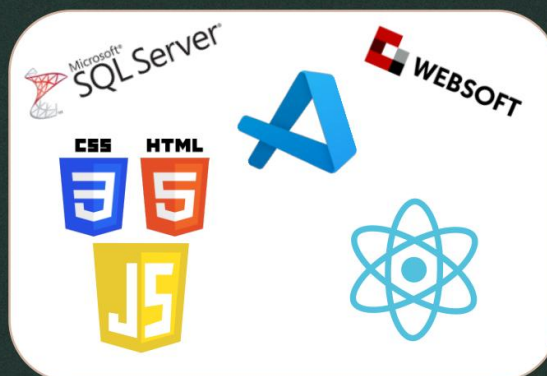
Изм.	Лист	№ докум.	Подпись	Дата

ПГУ 1.090501.12.001 ПЗ

Лист

141

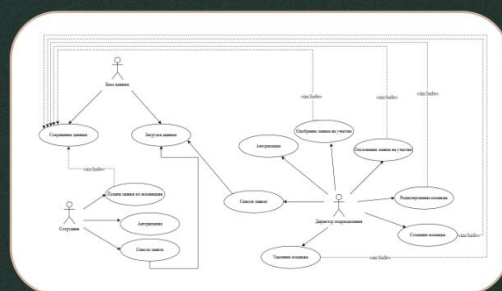
- Websoft HCM
- React, JavaScript, HTML/CSS
- MS SQL Server
- VS Code



АРХИТЕКТУРА МОДУЛЯ

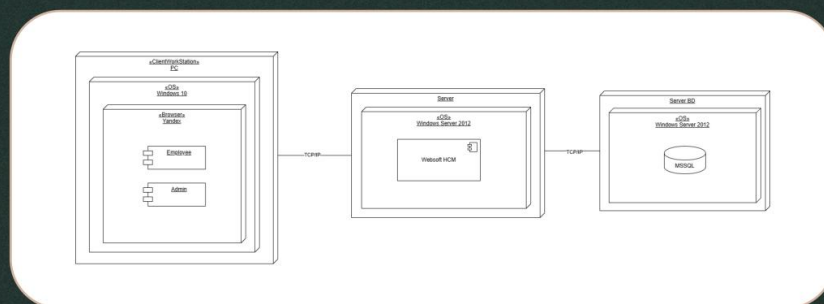
[illegible]

Этап регистрации



Этап проведения оценки

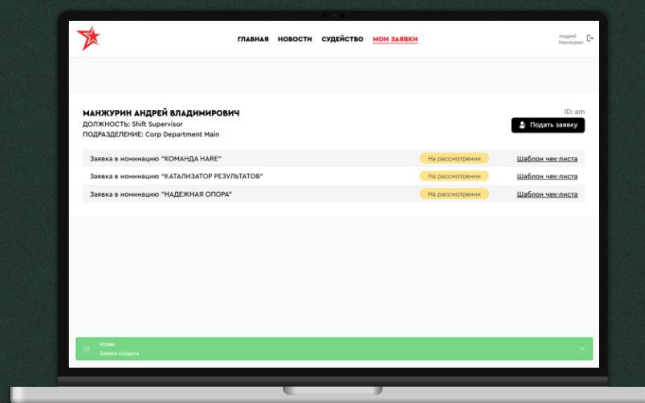
UML-ДИАГРАММА РАЗВЕРТЫВАНИЯ



РАЗРАБОТКА МОДУЛЯ



СТРАНИЦА ПОДАНЫХ ЗАЯВОК



Изм.	Лист	№ докум.	Подпись	Дата

ПГУ 1.090501.12.001 ПЗ

Лист

143

СТРАНИЦА СУДЕЙСТВО

Этап регистрации



СТРАНИЦА СУДЕЙСТВО

Этап оценки

СТРАНИЦА УПРАВЛЕНИЯ КОМАНДОЙ



ЗАКЛЮЧЕНИЕ

Разработан модуль, решающий задачу автоматизации оценки персонала.

Повышена эффективность HR-процессов.

Возможности расширения:

- геймификация;
- расширенная аналитика;
- AI-анализ компетенций.

СПАСИБО ЗА ВНИМАНИЕ !

					ПГУ 1.090501.12.001 ПЗ	Лист
						145
Изм.	Лист	№ докум.	Подпись	Дата		