

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное бюджетное образовательное учреждение высшего образования

«ПЕНЗЕНСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»

Факультет

Кафедра

вычислительной техники

«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

ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ ОТДЕЛА КАДРОВ

Студент Комолов 14.06.2024 Комолов Роман Игоревич
(подпись, дата) (ФИО полностью)

Руководитель Федюнин 14.06.2024 Федюнин Р.Н.
(подпись, дата) (фамилия, инициалы)

Нормоконтролёр Бычков 14.06.2024 Бычков А.С.
(подпись, дата) (фамилия, инициалы)

Рецензент зав. каф. Программирования 14.06.2024 Мартышкин А.И.
(должность, место работы) (подпись, дата) (фамилия, инициалы)

Работа допущена к защите (протокол зас. кафедры от 17.05.2024 г. № 11)

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

Работа защищена с отметкой удовлетворительно
(протокол заседания ГЭК от 19.06.2024 № 7)

Секретарь ГЭК

(подпись)

Гурин Е.И.
(фамилия, инициалы)

Пенза 2024

Федеральное государственное бюджетное образовательное учреждение
высшего образования

ПЕНЗЕНСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

ПОЛИТЕХНИЧЕСКИЙ ИНСТИТУТ

Факультет

Кафедра

Вычислительной техники

Вычислительная техника

«Утверждаю»
Заведующий кафедрой


Митрохин М.А.
«19» апреля 2024 г.

ЗАДАНИЕ НА ДИПЛОМНЫЙ ПРОЕКТ

1. Студент Комолов Роман Игоревич гр. 19ВОЭ1 факультета
вычислительной техники.

2. Тема работы: «Программное обеспечение отдела кадров».

Тема утверждена приказом ПГУ № 405/0 от 19.04.2024 г.

3. Руководитель работы к.т.н., доцент каф. ВТ «ПГУ» Федюнин Р.Н.

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

Разработать программное приложение для кадрового контроля командной
работы инженерных отделов IT организации.

Программное приложение должно выполнять следующие функции:

создание, добавление, удаление списка должностей, создание,
редактирование, удаление информации о персонале, формирование и
ведение информации о потребности сотрудников в отпусках и групповом
обсуждении проблем коллектива и проектов.

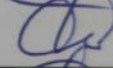
Программное обеспечение должно поддерживать много пользовательский
режим работы, разные типы пользователей должны иметь свой уникальный
набор функций.

Программное приложение должно разрабатываться с помощью среды
разработки Visual Studio Code. База данных должна разрабатываться для
СУБД реляционного типа.

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.

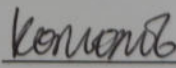
6. Календарный график выполнения работы

№ п/п	Наименование этапов работы	Объем работы	Срок выполнения	Подпись руководителя
1.	Анализ предметной области	10	19.04 - 24.04	
2.	Проектирование программного приложения	10	25.04 - 30.04	
3.	Разработка базы данных	20	01.05 - 06.05	
4.	Отладка и тестирование	10	07.05 - 13.05	
5.	Разработка экономической и части безопасности жизнедеятельности	10	14.05 - 28.05	
6.	Оформление графической части	10	29.05 - 01.06	
7.	Оформление выпускной работы	30	02.06 - 05.06	

Дата выдачи задания 19.04 » 2024 года

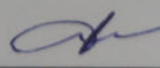
Руководитель дипломного проекта  Федюнин Р.Н.

(подпись, дата)

Задание к исполнению принял студент  Комолов Р.И.

(подпись, дата)

Работу к защите допустить
Декан факультета



Фионова Л.Р.

17.06.2024

(подпись, дата)

Реферат

Пояснительная записка 118 листов, 32 рисунка, 21 таблица и 3 приложения

ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ, ПЕРСОНАЛ, КОМАНДА СОТРУДНИКОВ, БАЗА ДАННЫХ, WEB-АРХИТЕКТУРА

Объектом исследования является процесс управления персоналом команд IT-организаций, с участием сотрудников отдела кадров и руководителей команд.

Цель работы – разработка программного приложения для помощи сотрудникам IT-команды с целью их профессионального развития и максимального удержания в рамках организации «мягкими» способами.

В процессе работы необходимо выполнить анализ предметной области, сформулировать техническое задание для разрабатываемого программного приложения, разработать логическую и физическую модели данных для разработки базы данных.

В результате проделанной работы разработано программное приложение, состоящее из базы данных и веб-приложения.

База данных спроектирована с использованием CASE-средства *StartUML*. База данных разработана для СУБД *MySQL*, приложение реализовано в среде разработки *Visual Studio Code*.

<h2 style="margin: 0;">ПГУ 1.090501.05.001 ПЗ</h2>				
Изм.	Лист	№ докум.	Подпись	Дата
Разраб.		Комолов Р.И.		14.06.24
Провер.		Федюнин Р.Н.		14.06.24
Н. контр.		Бычков А.С.		14.06.24
Утв.		Митрохин М.А.		14.06.24
Программное обеспечение отдела кадров Пояснительная записка				
ФВТ, каф. ВТ гр.19В031				

Содержание

Перечень сокращений и обозначений	4
Введение	5
1 Анализ предметной области	7
1.1 Описание предметной области	7
1.2 Анализ существующих аналогов	10
1.3 Цели и задачи разрабатываемой программы	13
1.4 Анализ целевой аудитории	13
2 Техническое задание	14
2.1 Назначение разработки	14
2.2 Требования к программе	14
2.3 Требования к программной документации	15
2.4 Стадии и этапы разработки	16
2.5 Порядок контроля и приемки	17
3 Обоснование выбора инструментов разработки	18
3.1 Язык сценариев JavaScript	18
3.2 Язык разметки HTML	21
3.3 Язык стилей CSS	23
3.4 Выбор языка программирования	24
3.5 Выбор CASE-средства	26
3.6 СУБД MySQL	28
4 Разработка программного обеспечения отдела кадров	32
4.1 Проектирование базы данных приложения	32
4.2 Реализация программного приложения	40
4.3 Реализация интерфейса программы	47
5 Оценка затрат на разработку и внедрение программы для интерактивного тестирования IT специалистов	59

6 Общие принципы обеспечения безопасности на рабочем месте оператора	
программного приложения	67
Заключение	71
Список использованных источников	72
Приложение А – UML-диаграммы	73
Приложение Б – Листинг программы	81
Приложение В – Презентация	101

Перечень сокращений и обозначений

В настоящей работе применяют следующие сокращения и обозначения

БД – база данных

FK (Foreign Key) – внешний ключ

ID (Identifier) – идентификатор объекта

PK (Primary Key) – первичный ключ

SQL (Structured Query Language) – язык структурированных запросов

UML (Unified Modeling Language) – унифицированный язык моделирования

HRIS – информационная система работы с персоналом

HRMS – система управления персоналом

					<i>ПГУ 1.090501.05.001 ПЗ</i>	Лист
						5
Изм.	Лист	№ докум.	Подпись	Дата		

Введение

Количество IT-организаций различного уровня и направления растет в Российской Федерации каждый год на 10% [1].

С возникновением новых IT-компаний и команд возникает потребность в кадровом управлении сотрудниками и процессами, которые с ростом количества сотрудников возникают: начисление заработной платы, бюрократические моменты с приказами и служебными записками, отпуска, формирование команд, повышение квалификации и так далее.

Для решения данных кадровых (далее - HR) вопросов в организации формируют отделы кадров и закупают специализированное программное обеспечение, несмотря на это, существуют области управления персоналом которые стандартные решения «закрывают» частично [2].

В частности, задачи фасилитации [2], подбора сотрудников в команду, психологический настрой команды или ее отдельных членов не учитываются только цифрами и зачастую не фиксируются стандартным HR программным обеспечением.

В результате в разгар проекта разработчик увольняется без объяснения причин, сотрудники отдела кадров и лидер команды получают задачу найти нового сотрудника и ввести его в курс дела, а этот процесс занимает немало времени. В результате и бизнес и команды теряют прибыль и психологический настрой на работу, а на решение «пожарных» вопросов тратится много рабочего времени.

В данной дипломной работе реализовано программное обеспечение, которое является отдельным рабочим модулем, который используется параллельно стандартным решения, таки как 1С Кадры[3] или Directum HR [4].

Цель реализованного программного обеспечения связать разработчиков, лидера команды и сотрудника отдела кадров общим информационным пространством, для решения следующих задач:

– возможность запроса отпуска сотрудником;

					<i>ПГУ 1.090501.05.001 ПЗ</i>	Лист
						6
Изм.	Лист	№ докум.	Подпись	Дата		

- возможность запроса на фасилитацию сотрудником с целью обсуждения командной или личной проблемы;
- возможность лидеру команды видеть настрой и возможности подчиненных ему команд на основании запросов;
- возможность сотруднику отдела кадров добавлять новые командные роли и поддерживать связь с лидером команды и отдельными сотрудниками.

Наличие отдельного модуля позволит не перегружать его стандартными бюрократическими процедурами, исключить использование в нем персональных данных [5] и включить в совместную работу: сотрудника отдела кадров, тимлида, сотрудника группы психологической поддержки, фасилитатора IT-команд и каждого члена команды.

Сотрудники команд в свою очередь получают дополнительный инструмент, через который они смогут получать информацию о своей карьерной траектории и рабочем времени, отпусках и возможности довести проблематику командной работы всем заинтересованным лицам в компании.

Программное приложение реализовано на базе трехслойной архитектуры и включает в себя базу данных и программное веб-приложение. Клиентская часть организована в виде «тонкого» клиента.

При проектировании базы данных использовалось CASE-приложение, для реализации приложения была применена среда разработки Visual Studio Code.

Все поставленные в дипломном проекте задачи были реализованы.

1 Анализ предметной области

1.1 Описание предметной области

В настоящий момент существует широкий спектр программного обеспечения для информационной поддержки отдела кадров. Типичное программное обеспечение для специалиста по персоналу выполняет следующие функции:

- хранение и систематизация информации о кандидатах и сотрудниках на конкретных позициях;
- планирование управления расчетом заработной платы;
- совершенствование процесса привлечения кандидатов в компанию;
- повышение удержания и вовлеченности сотрудников;
- предоставление возможностей и отслеживание прогресса людей (тренинги и повышение квалификации);
- запуск отчетов и подготовка KPI для аналитики и отчетности (внутренняя и внешняя);
- отслеживание рабочего времени, посещаемости, согласование с расчетом заработной платы (время - деньги);
- улучшение кадрового потенциала и планирование преемственности;
- мониторинг прогресса и обзоры эффективности управления сотрудниками.

Цифровизация различных отраслей экономики [7], в том числе вопросов кадрового обеспечения и документооборота привели к тому, что появились программные решения для отделов кадров специализированных предприятий, в частности IT-сектора [8].

Типичное приложение для отдела кадров является мультипользовательским приложением, в котором логин, пароль и роль пользователя определяют набор функций ПО который ему доступен.

					<i>ПГУ 1.090501.05.001 ПЗ</i>	Лист
						8
Изм.	Лист	№ докум.	Подпись	Дата		

Основные роли доступные в типичном корпоративном приложении – сотрудник отдела кадров, руководитель отдела кадров, представитель подразделения.

Сотрудники отдела кадров следят за приказами, заработной платой и иной бюрократической рутинной. Представитель подразделения в лице руководителя или доверенного лица участвует в мониторинге кадрового процесса и формировании документов.

Но за всем этим корпоративные решения зачастую «теряют» личность сотрудника и как результат команда получает сигнал об уходе сотрудника, отсутствии у него мотивации или квалификации в самый неподходящий момент.

В рамках дипломного проекта была поставлена задача - разработать программное обеспечение, которое будет дополнять стандартные внедренные в организации решения кадрового управления.

Первым этапом разработки ПО стало описание ролей и функций пользователей разрабатываемого приложения с помощью диаграммы вариантов использования.

На диаграмме вариантов использования (рисунок 1) показаны роли и действия позволяющие программному обеспечению вовлечь в общий рабочий процесс и единое информационно пространство отдел кадров, руководителей команд разработчиков и сотрудников команд разработки.

Также в ПО присутствуют «классические» для данного программного продукта функции – работа с информацией о сотрудниках, запросы сотрудников на реализацию тех или иных действий относительно места работы.

Только в отличие от большинства имеющихся решений в ПО была добавлена функция запроса на фасилитацию со стороны сотрудника, возможность выбора команды и проекта сотрудником с утверждением или отклонением данного запроса лидером команд или сотрудником отдела кадров отделом.

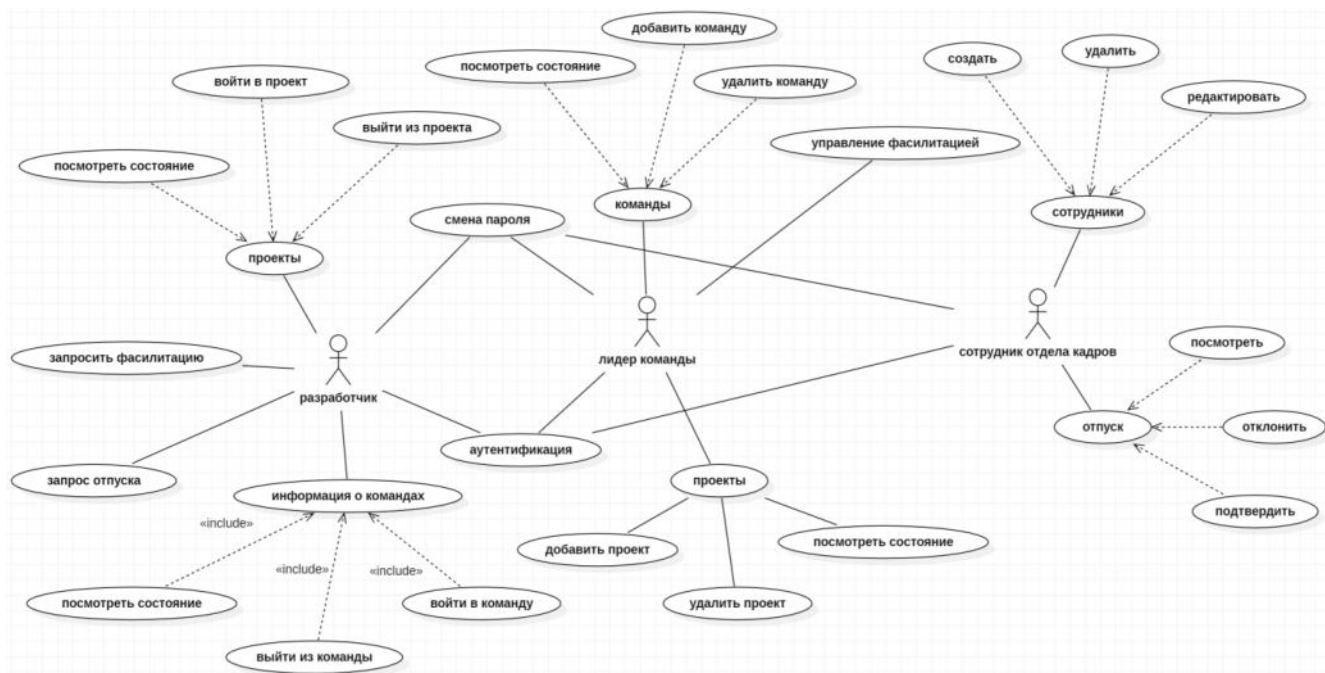


Рисунок 1 – Диаграмма вариантов использования

Данный функционал позволяет сделать следующие вещи:

- поддерживать уровень доверия в команде;
- выстроить диалог в цепочке: сотрудник команды, лидер команды, отдел кадров;
- иметь возможность довести информацию о проблемах или успехах в команде до максимального числа сотрудников.

Для достижения поставленных задач, в программе определено три вида ролей: разработчик, лидер команды и сотрудник отдела кадров.

Разработчик имеет возможность, авторизоваться в системе и получить информацию о командах и проектах, в которых он участвует.

При этом у пользователя разработчик есть возможность влиять на свою загруженность в командах и проектах путем пунктов меню: выйти из команды или выйти из проекта или войти в команду или войти в проект.

В организации это влияет на рейтинг и заработную плату сотрудника, а также влияет на его резюме.

Для решения вопросов проекта или команды разработчик может запросить сессию фасилитации, данный запрос будет отправлен лидеру команды и

сотруднику отдела кадров, которые при необходимости соберут митап, совещание или позовут стороннего фасилитатора, если вопрос в команде требует не только технических компетенций, но и помощи психолога и переговорщика.

Роль лидера команды имеет три основные функции: управление командами, управление проектами и коммуникация с сотрудниками и отделом кадров. Благодаря настроенным пунктам меню лидер команды может делать вывод о состоянии команд и делать запрос на подбор новых сотрудников в отдел кадров или запрос на фасилитацию команды.

Роль отдела кадров управлять человеческим ресурсом команд, регламентировать время отпуска и запросы на фасилитацию.

Описанные задачи на реализацию программного обеспечения в рамках дипломного проекта требуют анализа аналогов для определения целесообразности реализации проекта.

1.2 Анализ аналогов

В настоящий момент на международном рынке труда существуют следующие комплексные сервисы для автоматизации работы отдела кадров IT компании:

– BambooHR: облачный сервис, предназначенный для HR-специалистов и специалистов по подбору персонала, решающий основные задачи управления кадровыми данными на малых и средних предприятиях;

– Comindware: предлагает систему управления HR-процессами, делающую их более понятными и эффективными;

– LeaderTask: предоставляет ряд функций, включая учет времени, контроль посещаемости и редактирование документов, для упрощения работы отдела кадров во всех аспектах;

– PeopleForce: система, состоящая из различных инструментов для поддержки управления процессами управления персоналом или отдельных

инструментов для конкретных процессов, подчеркивающая важность удобной конфигурации и доступности;

– SAP Staff Central: комплексная HRIS (информационная система управления персоналом), которая охватывает практики, политики, процессы и программное обеспечение, используемые организациями для управления своей рабочей силой, обеспечивая соблюдение глобальных трудовых норм и поддерживая сотрудников;

– 1С Кадры предлагает широкий спектр функций для управления HR-процессами, включая расчет заработной платы, льгот, обучение и управление талантами.

Эти решения систем управления кадрами (HRMS) направлены на автоматизацию и оптимизацию различных HR-процессов, таких как подбор персонала, управление данными о сотрудниках, расчет заработной платы, льгот, обучение и управление талантами. Они предназначены для повышения эффективности, снижения затрат и улучшения общего опыта сотрудников.

Все ведущие комплексные программы для отдела кадров – импортные решения за исключением 1С Кадры. Более того это зачастую облачные решения как SAP, которые хранят всю информацию о предприятии на зарубежных серверах.

В связи с усилением мер в рамках Федерального закона № 152-ФЗ «О персональных данных», который устанавливает принципы и правила сбора, хранения, использования и распространения персональных данных, а также права и обязанности физических лиц и организаций, участвующих в обработке персональных данных, зарубежные решения прекратили свою деятельность в РФ. Но потребность в данных решениях осталась особенно в IT секторе [9].

Если сравнивать отечественные и зарубежные решения по функциональным характеристикам и стоимости, то можно прийти к следующему выводу.

Отечественные программные решения (в частности 1С Кадры) обладают следующими достоинствами:

– комплексное управление персоналом – наличие интеграции с продуктами 1С. Большинство ПО РФ имеет широкий спектр функций для управления HR-процессами и интеграцией с решениями 1С и Парус;

– наличие функций взаимной миграции;

– поддержка русского языка, так как ПО разработано специально для российского рынка, что делает ее отличным выбором для организаций, работающих в России.

Но есть и существенные минусы данного вида отечественного ПО:

– крутая кривая обучения для использования и доработки ПО требует дорогостоящего обучения для эффективного использования;

– ограниченные возможности настройки. Пользователи сообщают об ограниченных возможностях настройки, что может затруднить адаптацию системы к их конкретным потребностям.

В тоже время западные аналоги:

– зачастую лишены недостатков, связанных с поддержкой и интеграцией ПО;

– имеют интуитивно понятный и удобный интерфейс, который упрощает навигацию и использование специалистами разного уровня;

– ПО предлагает высокую степень настройки, позволяя организациям адаптировать систему к своим конкретным потребностям;

– комплексные функции: ПО предлагает широкий спектр функций, включая подбор персонала, управление данными о сотрудниках, расчет заработной платы, льготы, обучение и управление талантами.

Исходя из выше сказанного можно сделать вывод, что реализация отечественного программного обеспечения, которое будет открыто, бесплатно и имеет необходимые функции для управления кадрами в рамках ИТ организации является перспективным направлением.

При этом фактически все отечественные решения лишены функций работы с психологическим фактором работы в команде и фасилитацией команд.

Вышесказанное требует поставить цель и задачи, решаемые в рамках дипломного проекта.

1.3 Цели и задачи разрабатываемого программного приложения

Цели реализации дополнительного программного обеспечения отдела кадров:

- увеличение скорости распространения информации в организации от сотрудника руководству и отделу кадров;
- обеспечение возможности сотруднику влиять на свой отпуск посредством запроса и обсуждения;
- структуризация деятельности команд;
- структуризация деятельности проектов.

В результате внедрения программного приложения планируется достичь ряда задач, а именно:

- повысить эффективность работы сотрудников;
- снизить напряженность сотрудников и повысить коммуникацию в организации.

1.4 Анализ целевой аудитории

Так как команды разработчиков в IT компаниях могут работать удаленно и часть разработчиков могут быть интровертами, то наличие отдельного программного модуля для элементарного оповещения лидера команды, коллег и отдела кадров необходима IT-компаниям. Продуктов-аналогов не было найдено. Как следствие, разработка программного приложения является актуальной как для частных, так и для государственных компаний, у которых есть командная IT работа.

2 Техническое задание

2.1 Назначение разработки

Программное обеспечение отдела кадров предназначено для автоматизации работы с персоналом команд IT-компаний.

Архитектурные задачи данного проекта – разработка базы данных и веб-приложения.

2.2 Требования к программе

2.2.1 Требования к функциональным характеристикам

Программное приложение контроля выполнения задач должно обеспечивать выполнение следующих функций:

- авторизация пользователей;
- регистрация пользователей;
- удаление, редактирование и добавление информации о сотрудниках;
- формирование запроса от сотрудников на отпуск и фасилитацию;
- формирование запроса от руководства подразделениями в отдел кадров;
- информирование сотрудника о статусе его запросов.

Входными данными для системы должны быть:

- сведения о сотрудниках;
- сведения о потребности сотрудника;
- сведения о потребности подразделения.

2.2.2 Требования к надежности

Приложение должно работать без прерываний и ошибок. В случае если пользователь совершает неверные действия, система должна уведомить его об этом.

Программное приложение должно обеспечивать контроль вводимых данных. В случае ввода некорректных данных, или не ввода данных совсем приложение должно информировать об этом пользователя.

					<i>ПГУ 1.090501.05.001 ПЗ</i>	Лист
						15
Изм.	Лист	№ докум.	Подпись	Дата		

При вводе входной информации должен осуществляться ее контроль на допустимость вводимых значений по следующим правилам:

- вводимые данные не должны противоречить заявленному типу данных;
- вводимые данные не должны противоречить законам логики.

При выполнении операции по изменению и удалению данных должна быть обеспечена целостность базы данных.

Конечный пользователь не должен иметь доступ к исходным данным приложения. Для модификации программной части системы пользователю необходимо обратиться к разработчику программного приложения.

2.2.3 Требования к составу и параметрам технических средств

Разрабатываемое приложение должно работать в условиях локальной сети предприятия.

Программное приложение предназначено для работы на персональных компьютерах сотрудников ИТ-компании.

Минимальная конфигурация технических средств должна включать:

- ОС Windows 8 и выше;
- процессор Intel Core или аналог;
- свободное место на жестком диске 500 Мб и выше;
- оперативную память 8 ГБ и выше;
- монитор с диагональю не менее 16 дюймов;
- клавиатура;
- мышь.

2.3 Требования к программной документации

Программное приложение должно сопровождаться документацией, содержащей:

- техническое задание;
- описание логического проектирования БД;
- описание программы.

2.4 Стадии и этапы разработки

Этапы разработки программного приложения представлены в таблице 1.

Таблица 1 — Стадии и этапы разработки

Стадии	Этапы	Объем работы в процентах	Срок выполнения	Исполнитель
1 Анализ предметной области	Анализ предметной области	10		Комолов Р.И.
2 Проектирование программного приложения	-концептуальное проектирование БД -логическое проектирование БД -физическое проектирование БД	10		Комолов Р.И.
3 Разработка базы данных	Разработка базы данных	20		Комолов Р.И.
4 Отладка и тестирование	Отладка и тестирование	10		Комолов Р.И.
5 Разработка экономической части	Оценка затрат на разработку и внедрение программного приложения контроля выполнения задач	5		Комолов Р.И.

Стадии	Этапы	Объем работы в процентах	Срок выполнения	Исполнитель
6 Разработка раздела БЖД	Оценка тяжести труда эксплуататора программного приложения	5		Комолов Р.И.
7 Оформление графической части	Оформление графической части	10		Комолов Р.И..
8 Оформление Выпускной работы	Оформление пояснительной записки	30		Комолов Р.И.

2.6 Порядок контроля и приемки

Для проверки выполнения программой всех ее функций в соответствии с требованием технического задания необходимо разработать тестовые наборы данных. Необходимо протестировать действия программы на возможные реакции пользователя: запуск программы, заполнение полей ввода, выбор пунктов меню, нажатие кнопок, работа с базой данных. Приложение соответствует требованиям технического задания, если оно выполняет все заданные функции при различных входных данных из тестовых наборов данных.

Проверка и приемка программы должна осуществляться при ее работоспособности и в присутствии заказчика, выполнения всех возложенных на нее функциональных возможностей, а также при наличии полной документации на программу (пояснительная записка с приложениями: текстом программы и результатом текстовых примеров).

3 Обоснование использования инструментов разработки

На основании анализа требований к функциональным характеристикам программного приложения были выбраны наиболее актуальные средства разработки.

Для создания WEB-страниц, с элементами форматирования и интерактивности используется классический стек программирования интерфейса: HTML, CSS, JavaScript.

Для реализации логики приложения и работы с данными используется связка языка PHP и СУБД MySQL.

Взаимодействие клиента с приложением осуществляется по протоколу HTTP и с использованием архитектуры «тонкого» клиента.

Ниже представлены разделы с описанием обоснования выбора стека технологий для реализации программного обеспечения отдела кадров IT-компании.

3.1 Язык сценариев JavaScript

JavaScript язык программирования (рисунок 2), который широко используется в WEB-разработке для добавления интерактивности и динамичности на WEB-страницах [5]. JavaScript имеет как преимущества, так и недостатки, а также имеет аналоги.

JavaScript, TypeScript и CoffeeScript - три популярных языка сценариев [6], используемых в WEB-разработке. Ниже представлено сравнение особенностей, преимуществ и недостатков каждого из данных языков сценария.

Функции JavaScript: динамическая типизация, объектно-ориентированное программирование на основе прототипов, поддержка событийно-ориентированного и функционального стилей программирования.

Преимущества JavaScript: широкое использование в WEB-разработке, простой порог вхождения в разработку на данном языке, высокий спрос на рынке труда [7].

Недостатки JavaScript: динамическая типизация может привести к ошибкам, нет встроенной поддержки проверки типов или статической типизации.

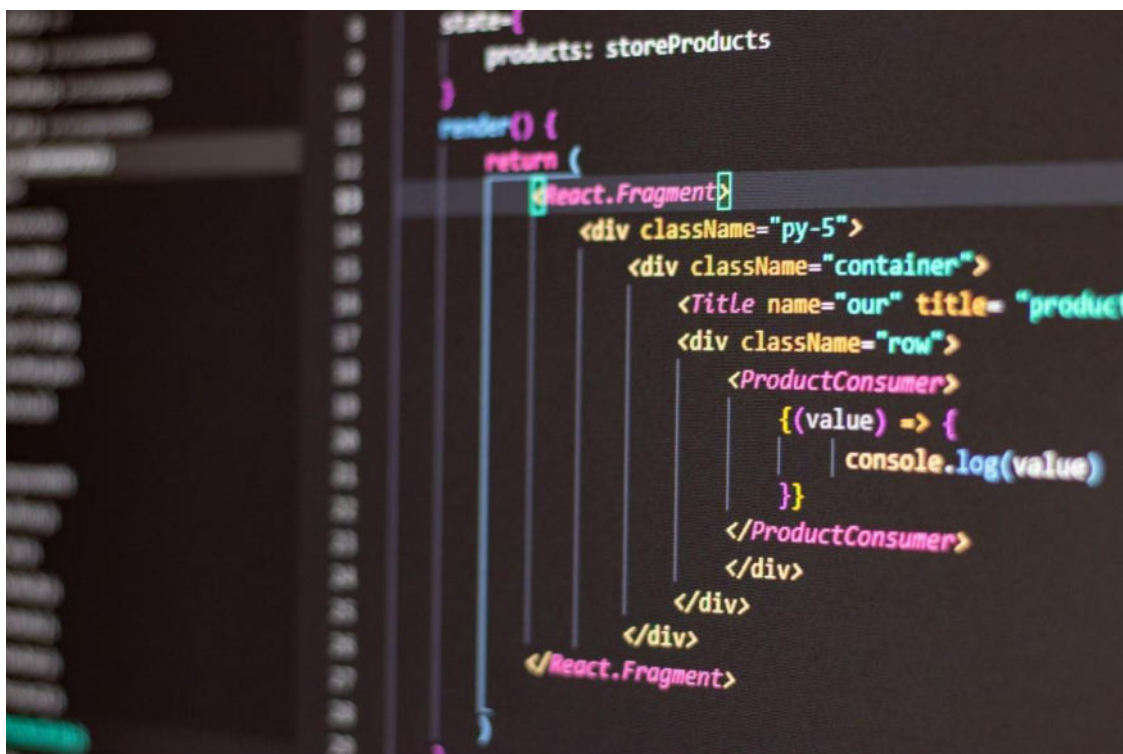


Рисунок 2 – Язык программирования JavaScript

TypeScript - статически типизированная расширенная версия JavaScript, разработанная Microsoft [6] (рисунок 3).

Функции TypeScript: статически типизированный, поддерживает проверку типов и статическую типизацию, совместим с кодом JavaScript.

Имеет встроенный функционал ООП: функции, классы, модули и интерфейсы.

Преимущества TypeScript: встроенные продвинутые средства сопровождения и читаемости кода, встроенные библиотеки обнаружения и предотвращения ошибок, поддерживает крупномасштабные приложения, имеет совместимость с Angular и другими фреймворками.

Недостатки TypeScript: крутая кривая обучения из-за статической типизации, лишней для небольших проектов.

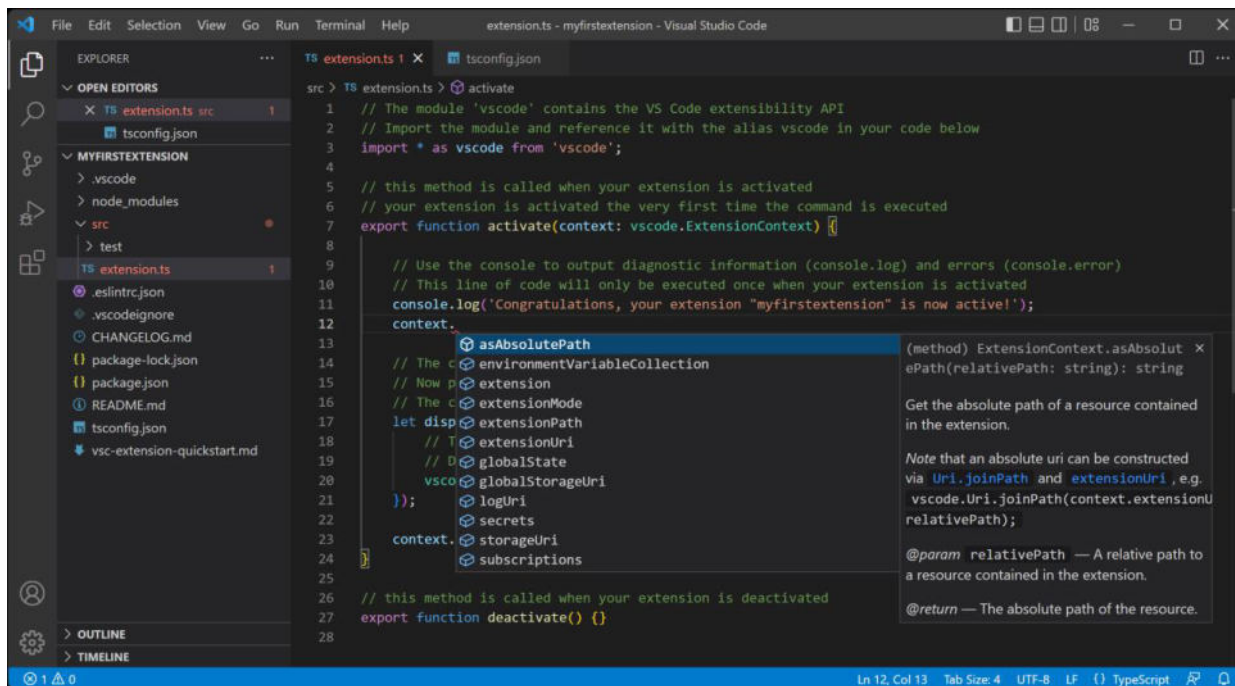


Рисунок 3 – Язык программирования TypeScript

CoffeeScript - это динамичный язык сценариев, который компилируется в JavaScript (рисунок 4).

Функции CoffeeScript: упрощенный синтаксис и улучшенная читаемость по сравнению с TypeScript. Компилируется в JavaScript.

Преимущества CoffeeScript: Содержит встроенную поддержку Ruby и Python. Удобный и компактный язык для средних и больших проектов.

Недостатки CoffeeScript: ограниченная поддержка сообщества и разработчиков языка, не так широко используется, как JavaScript или TypeScript. CoffeeScript чувствителен к пробелам и иным «нечитаемым» символам.

В связи с тем, что реализуемое приложение относится к классу минимально продукта (MVP) JavaScript будет оптимальным выбором для реализации динамики интерфейсов приложения. JavaScript обеспечит простоту разработки и удобочитаемости кода приложения, поддержку и автоматическое обнаружение ошибок.

```

demo.coffee
4
5 # Conditions:
6 number = -42 unless opposite
7
8 # Functions:
9 • square = (x -> x * x
10
11 # Arrays:
12 list = [1, 2, 3, 4, 5]
13
14 # Objects:
15 math =
16   root: Math.sqrt
17   square: square
18   cube: (x) -> x * square x
19
Build Error missing )
CoffeeScript File 1 Project 1 1 issue demo.coffee 9:10 LF UTF-8 CoffeeScript

```

Рисунок 4 – Язык программирования CoffeeScript

3.2 Язык разметки HTML

HTML (Hypertext Markup Language) - язык разметки (рисунок 5), используемый для создания WEB-страниц. HTML имеет характеристики, которые будут рассмотрены ниже.

Характеристики HTML:

- структура: HTML используется для создания структуры WEB-страницы, включая заголовки, параграфы, изображения, ссылки и другие элементы;
- маркировка: HTML использует маркировку для определения различных элементов на WEB-странице;
- кросс-браузерность: HTML поддерживается большинством браузеров, что обеспечивает кросс-браузерную совместимость;
- легкость изучения: HTML прост в изучении, что делает его доступным для разработчиков.

```

index.html U x
index.html > html > head > title
1  <!DOCTYPE html>
2  <html>
3    <head>
4      <meta charset="utf-8" />
5      <meta name="viewport" content="width=device-width, initial-scale=1" />
6      <title>Списки</title>
7    </head>
8    <body>
9      <h2>Маркированный список</h2>
10
11      <ul>
12        <li>Элемент 1</li>
13        <li>Элемент 2</li>
14        <li>Элемент 3</li>
15      </ul>
16
17      <h2>Нумерованный список</h2>
18
19      <ol>
20        <li>Элемент 1</li>
21        <li>Элемент 2</li>
22        <li>Элемент 3</li>
23      </ol>
24    </body>
25  </html>

```

Маркированный список

- Элемент 1
- Элемент 2
- Элемент 3

Нумерованный список

1. Элемент 1
2. Элемент 2
3. Элемент 3

Рисунок 5 – Язык разметки HTML

Плюсы HTML:

- универсальность: HTML является стандартным языком разметки для WEB-страниц, что обеспечивает его универсальность;
- свободное распространение: HTML является бесплатным языком, что делает его доступным для всех разработчиков;
- взаимодействие с другими языками: HTML интегрируется с другими языками, такими как CSS и JavaScript.

Минусы HTML:

- статичность: HTML является статическим языком, что означает, что он не может создавать динамические интерактивные элементы без дополнительных языков, таких как JavaScript;
- ограничения безопасности: HTML имеет ограничения в плане безопасности, что делает его уязвимым для атак, таких как XSS и SQL-инъекции;
- сложность структуры: Создание и поддержка структуры HTML-документов может быть сложным, особенно для крупных проектов;

– длина кода: Создание WEB-страницы с помощью чистого HTML может усложнить его поддержку и обслуживание.

HTML - это мощный язык разметки, который применен в данном проекте для реализации интерфейса.

3.3 Язык стилей CSS

CSS (Cascading Style Sheets) - язык, используемый для описания внешнего вида и форматирования документов, написанных на языках разметки, таких как HTML или XML. CSS используется для управления внешним видом и поведением WEB-страниц, включая форматирование текста, цвета, шрифты, размеры и расположение элементов.

CSS состоит из селекторов, свойств и значений, которые определяют, как элементы на WEB-странице должны быть отображены. CSS используется для создания внешнего вида и форматирования WEB-страниц. CSS включает следующие функции:

- форматирование текста (шрифт, размер, цвет);
- создание макета страницы (расположение элементов, размеры);
- добавление декоративных элементов (анимации, эффекты).

Плюсы CSS:

- CSS прост в использовании и может быть легко интегрирован с HTML и JavaScript;
- CSS позволяет создавать стили для нескольких элементов на веб-странице, что уменьшает количество кода и упрощает его поддержку;
- CSS может быть кеширован WEB-браузером, что ускоряет загрузку веб-страницы;
- CSS может быть использован для создания доступных веб-страниц, которые могут быть просмотрены людьми с ограниченными возможностями;

– CSS позволяет создавать различные стили для различных устройств и браузеров, что обеспечивает кросс-браузерную и кросс-платформенную совместимость.

Минусы CSS:

- CSS может быть сложным при масштабировании больших проектов;
- CSS может иметь проблемы с совместимостью между разными браузерами и устройствами;
- CSS может быть уязвим для атак, таких как CSS-инъекции;
- CSS имеет ограничения в плане создания сложных интерактивных элементов, что может потребовать использования JavaScript.

Рассмотренные функции CSS показывают его достоинство и необходимость использования в данном проекте.

3.4 Выбор языка программирования

Для реализации логики приложения выбран язык программирования PHP. Ниже опишем достоинства и недостатки инструмента, которые аргументировали его выбор для реализации проекта.

Преимущества языка программирования PHP включают в себя:

- простой и интуитивно понятный синтаксис: PHP имеет легкий для понимания синтаксис, что делает его доступным для начинающих разработчиков;
- кроссплатформенность и гибкость: PHP может работать на различных платформах и операционных системах, что обеспечивает его широкую применимость;
- масштабируемость: PHP может эффективно обрабатывать большие объемы данных и масштабироваться для крупных проектов;
- встраиваемость в HTML-документы: PHP может быть встроен в HTML-код;

– бесплатность и открытый код: РНР является бесплатным языком с открытым исходным кодом, что делает его доступным для разработчиков и позволяет им вносить изменения;

– множество баз данных, фреймворков и библиотек: РНР имеет обширную экосистему инструментов и ресурсов, что облегчает разработку и поддержку проектов;

– динамическая типизация: РНР позволяет присваивать переменным различные типы данных, что обеспечивает гибкость в разработке;

– интерпретируемость: РНР не требует компиляции, что ускоряет процесс разработки и тестирования;

– широкая применимость: РНР используется в различных областях, включая веб-разработку, создание социальных сетей и блогов.

Недостатки языка РНР включают в себя:

– низкая степень защищенности: РНР имеет сравнительно низкую степень защищенности по сравнению с другими языками программирования, что может привести к проблемам с безопасностью;

– проблемы с глобальными исключениями: РНР может иметь трудности с глобальными исключениями, что может усложнить обработку ошибок;

– несогласованный стиль наименования и поведения параметров: Многие функции языка имеют несогласованный стиль наименования и поведения параметров, что может вызвать путаницу у разработчиков;

– ограничения в масштабируемости: РНР может иметь ограничения в масштабируемости для сложных проектов, что может быть проблемой для крупных приложений;

– асинхронность и современные технологии: РНР может не быть идеальным выбором для проектов, требующих асинхронности и использования современных технологий;

– производительность и скорость: РНР может иметь сравнительно низкую производительность и скорость по сравнению с другими языками программирования.

Несмотря на эти недостатки, РНР по-прежнему остается популярным выбором для веб-разработки из-за его простоты, гибкости, широкой поддержке и совместимости с многими базами данных.

3.5 Выбор CASE-средства

Для разработки базы данных и программного обеспечения используются специальные CASE-средства разработки концептуальных, логических и реляционных моделей. В данном проекте для проектирования ПО и БД используется StartUML (рисунок 6).

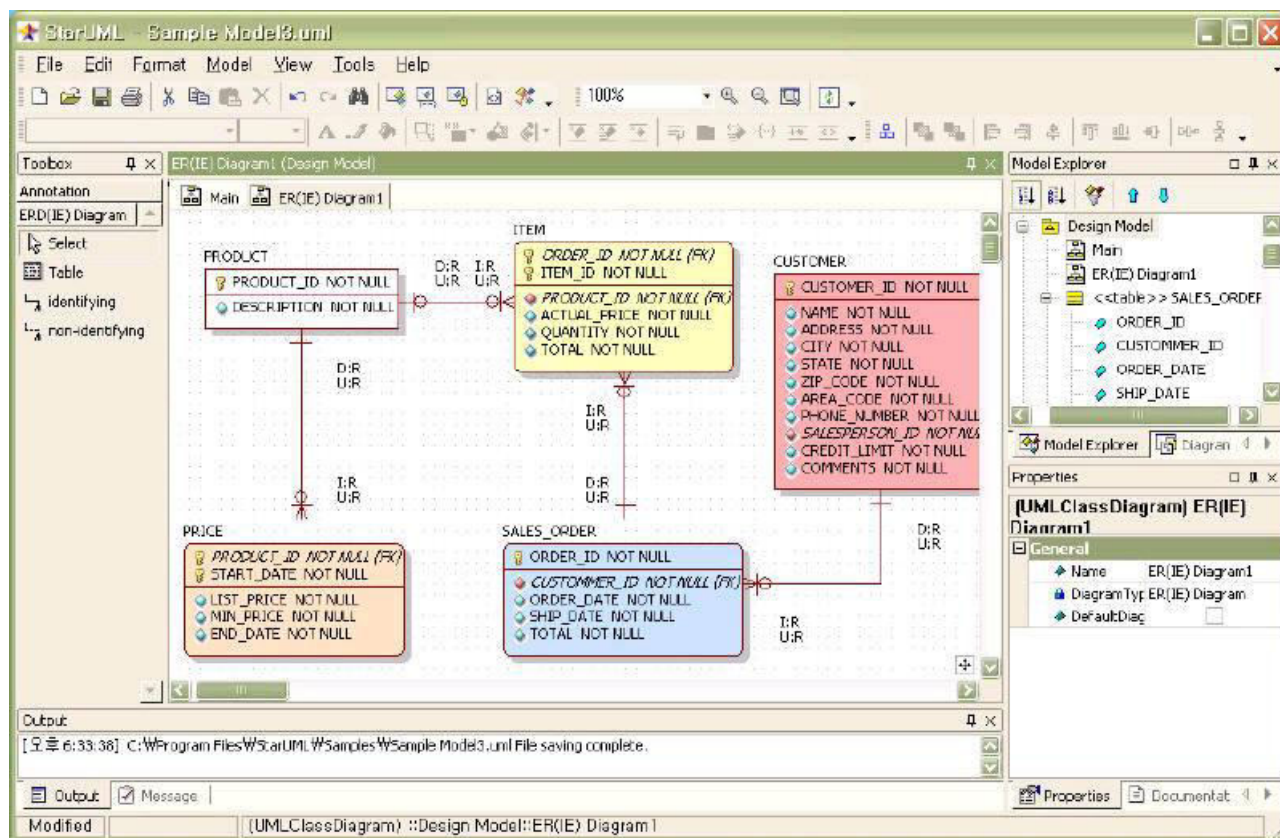


Рисунок 6 – Разработка БД в StartUML

StarUML – инструмент для моделирования, который поддерживает язык UML и другие нотации. Инструмент предлагает широкий спектр функциональных

возможностей, включая создание диаграмм, моделирование, кодогенерацию и интеграцию с другими инструментами.

Функциональные возможности StarUML:

– StarUML позволяет создавать различные типы диаграмм, включая классовые, компонентные, объектные, пакетные, диаграммы использования, диаграммы активности, последовательности, коммуникации, тайминга, состояния и информационного потока;

– инструмент может генерировать код в различных языках программирования;

– StarUML может быть интегрирован с другими инструментами, такими как Rational Rose, phpMyAdmin и прочие;

– инструмент обеспечивает управление моделями, включая создание, редактирование и управление моделями;

– StarUML позволяет настроить инструмент под свои нужды, включая создание пользовательских подходов и фреймворков;

– инструмент доступен на нескольких языках.

Плюсы StarUML:

– StarUML имеет интуитивно понятный интерфейс, что делает его простым в использовании;

– StarUML позволяет настроить инструмент под свои нужды, что делает его гибким и адаптируемым;

– StarUML является бесплатным инструментом, что делает его доступным для широкого круга пользователей.

Минусы StarUML:

– StarUML может быть сложным при работе с большими моделями;

– инструмент имеет ограничения в плане поддержки некоторых языков программирования и платформ одновременно;

– StarUML может иметь проблемы с совместимостью между разными версиями и платформами;

– инструмент может не иметь официальной поддержки для некоторых языков и платформ.

Вместе с тем, для реализации дипломного проекта StarUML обладает достаточным функционалом.

3.6 СУБД MySQL

Для функционирования базы данных необходимо выбрать систему управления базами данных.

СУБД (Система Управления Базами Данных) – комплекс программно-языковых средств, позволяющих создавать, управлять и администрировать базы данных. Она обеспечивает безопасность, надёжность хранения и целостность данных, а также предоставляет средства для администрирования БД.

СУБД необходимы для функционирования сайтов и приложений, так как они позволяют эффективно хранить и управлять данными.

Основные функции СУБД:

- управление данными: создание, изменение и удаление данных;
- управление доступом к данным для различных пользователей;
- защита от несанкционированного доступа.

Администрирование:

- обеспечение безопасности и целостности данных;
- резервное копирование и восстановление базы данных после сбоев;
- журнализация изменений.

MySQL, Oracle и Microsoft SQL Server - три популярные системы управления реляционными базами данных (СУБД) [7], используемые в различных отраслях. Вот сравнение этих систем:

Ключевые особенности СУБД Oracle:

- высокая надежность и масштабируемость;
- широкий ряд инструментов для хранения и анализа данных.

Минусы СУБД Oracle:

- высокие затраты на требования к оборудованию и лицензионные сборы;
- требуются дополнительные лицензии для многих функций.

Ключевые особенности Microsoft SQL-сервер:

- интегрирован с другими продуктами Microsoft;
- поддерживает как локальное, так и облачное развертывание;
- лаконичные инструменты резервного копирования и восстановления БД.

Минусы Microsoft SQL-сервер:

- дорогое корпоративное решение;
- использует язык T-SQL, имеющий отличия от стандарта SQL других

СУБД.

Ключевые особенности MySQL:

- открытый исходный код, что делает его экономически эффективным;
- высокая производительность, масштабируемость и удобство использования;
- поддерживает репликацию, полнотекстовый поиск и различные механизмы хранения;
- большая база пользователей и широкая поддержка.

Минусы MySQL:

- ограниченные возможности для сложной логики БД;
- проблемы стабильности и производительности в приложениях с высоким уровнем параллелизма.

При выборе системы управления реляционными базами данных (СУБД) для реализации приложения дипломного проекта учтены следующие факторы:

- стоимость. СУБД MySQL имеет открытый исходный код и экономически эффективен, в то время как Oracle и SQL Server являются дорогими корпоративными решениями.
- интеграция. SQL Server интегрируется с продуктами Microsoft, а MySQL широко поддерживается на всех платформах.

- масштабируемость. Все три варианта обеспечивают приемлемую масштабируемость для MVP.
- встроенный инструментарий разработки. Oracle и SQL Server предлагают более продвинутые функции для хранения данных и анализа, а MySQL подходит для WEB-приложений.
- поддержка производителя. MySQL имеет большую базу пользователей и широко поддерживается, тогда как Oracle и SQL Server требуют платы для полноценной технической поддержки.

3.7 Среда разработки Visual Studio Code

Для реализации проекта используется IDE Visual Studio Code – расширенный редактор кода, который используется для большого количества языков программирования. VS Code прост в освоении, подходит для начинающих разработчиков и мало весит. Обладает горячими клавишами, для использования функций без обращения к меню, а также в него встроен отладчик для, используемого при разработке приложения, языка JavaScript.

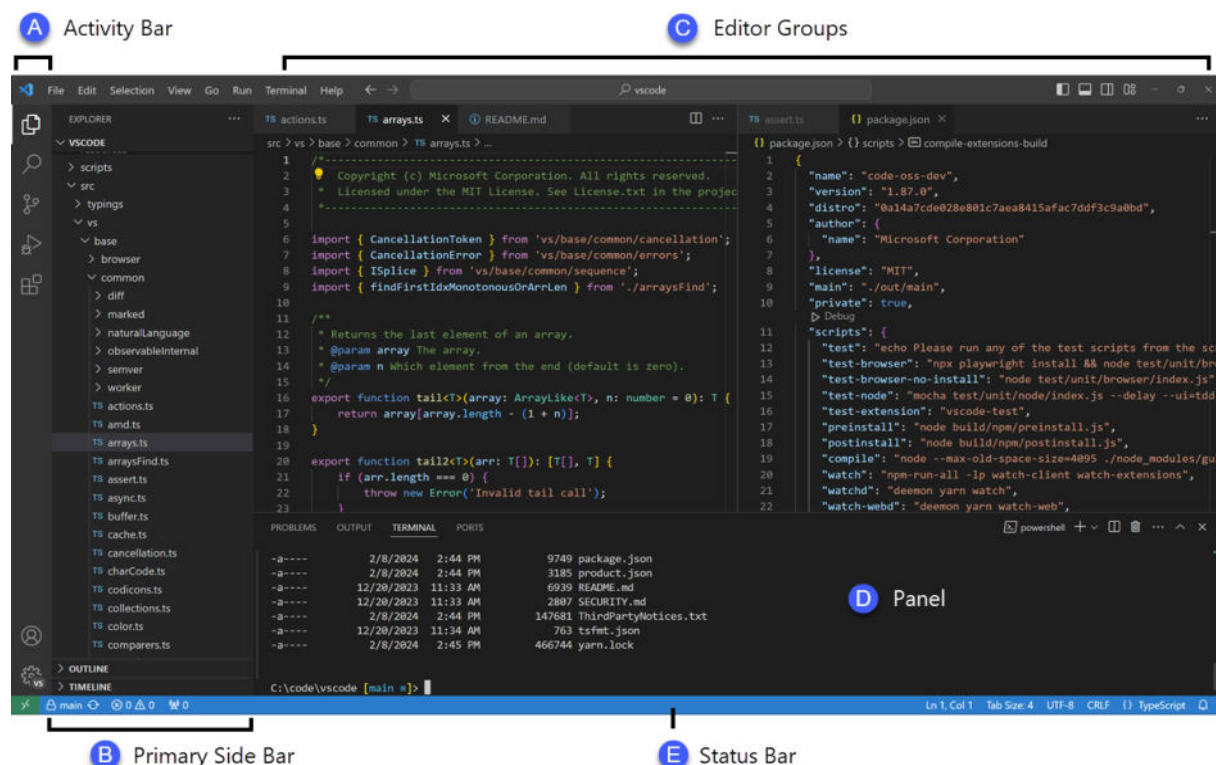


Рисунок 7 – Основное окно IDE Visual Studio Code

Изм.	Лист	№ докум.	Подпись	Дата

ПГУ 1.090501.05.001 ПЗ

Лист

31

Выбранный редактор кода является бесплатным и подходит для актуальных операционных систем.

Visual Studio Code обладает рядом преимуществ: легкость освоения, быстро работает и менее требователен в ресурсам компьютера, кроссплатформенность, поддержка большого количества языков программирования, гибкость, за счет множества расширений и удобных функций.

					<i>ПГУ 1.090501.05.001 ПЗ</i>	Лист
						32
Изм.	Лист	№ докум.	Подпись	Дата		

4 Разработка программного обеспечения отдела кадров

Реализованное программного обеспечения отдела кадров ИТ-компании состоит из трех основных частей: клиентская часть, серверная часть и система управления базой данных (рисунок 8).

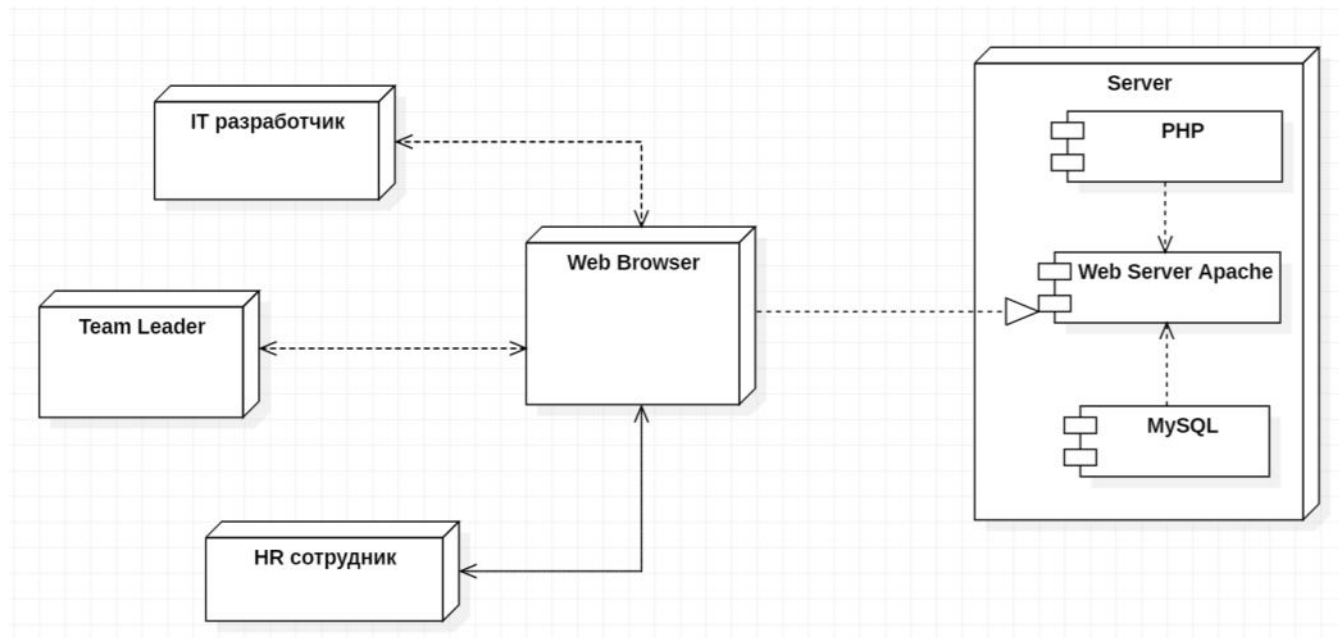


Рисунок 8 – Диаграмма развертывания приложения.

Как видно из диаграммы на рисунке 8 основной частью приложения является WEB сервер и СУБД MySQL. Клиенты приложения разделены на три роли, каждая из которых наделена уникальным функционалом.

Данные приложения хранятся в СУБД. Для реализации БД в следующем разделе описана схема сущностей и данных используемой СУБД.

4.1 Проектирование базы данных приложения

Логическая модель базы данных приведена на рисунке 9 и содержит 13 сущностей.

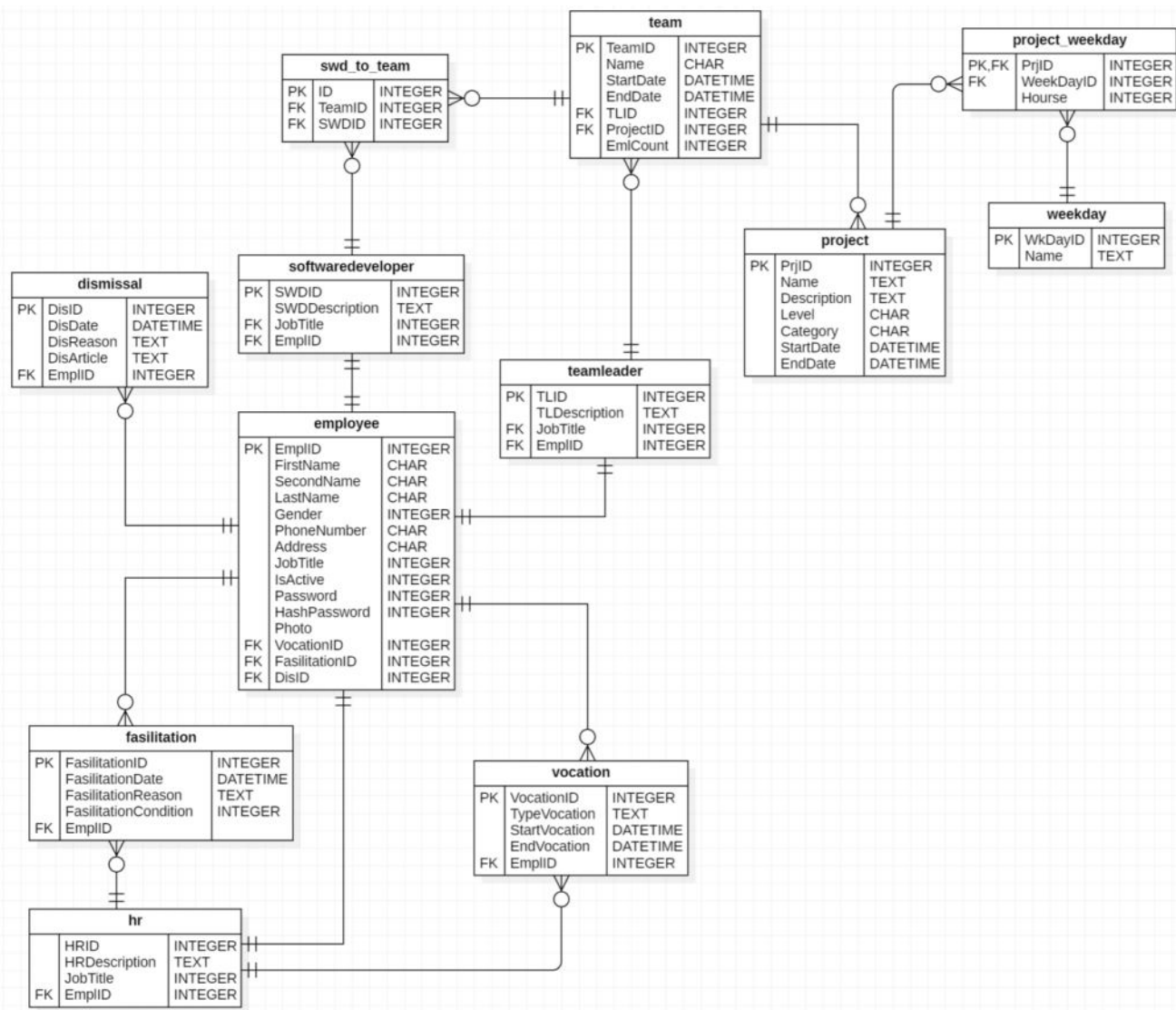


Рисунок 9 – Логическая модель данных

Прежде всего, БД должна позволять пользователям регистрироваться и входить в систему. Пользователем может быть сотрудник команды, лидер команды или сотрудник отдела кадров.

Сотрудники могут записаться в команду и на определенный проект. Сотрудники отдела кадров лидеры команд должны иметь возможность вводить информацию об участниках команд, подтверждать бронирование дат отпусков, запросов на фасилитацию и назначать учащихся в определенную команду. Кроме того, лидеры команд должны иметь возможность создавать новые команды и назначать в них сотрудников. Модель должна иметь возможность хранить подробную информацию о сотрудниках.

Ниже представлено подробное описание таблиц реализованной модели базы данных.

Таблица 2 – Описание атрибутов сущности «employee»

Имя атрибута	Назначение	Первичный ключ	Внешний ключ
EmplID	Уникальный идентификатор сотрудника	РК	
FirstName	Имя сотрудника		
SecondName	Отчество сотрудника		
LastName	Фамилия сотрудника		
Gender	пол (мужской/женский)		
PhoneNumber	Телефонный номер		
Address	Адрес сотрудника в формате город, улица, дом, квартира		
JobTitle	Наименование позиции в компании		
IsActive	Признак активности сотрудника		
Password	Пароль для входа в приложении		
HashPassword	Хеш пароля		
Photo	Ссылка на фотографию сотрудника		
VocationID	Ссылка на информацию об отпусках		FK
FasilitationID	Ссылка на информацию о сессиях фасилитации		FK
DisID	Ссылка на информацию об увольнении		FK

Таблица 3 – Описание атрибутов сущности «dismissal»

Имя атрибута	Назначение	Первичный ключ	Внешний ключ
DisID	Уникальный идентификатор процесса увольнения сотрудника	РК	
DisDate	Дата увольнения сотрудника		
DisReason	Причина увольнения сотрудника		
DisArticle	Статья закона как основание увольнения сотрудника		
EmplID	Уникальный идентификатор сотрудника		FK

Таблица 4 – Описание атрибутов сущности «fasilitation»

Имя атрибута	Назначение	Первичный ключ	Внешний ключ
FasilitationID	Уникальный идентификатор сессии фасилитации	РК	
FasilitationDate	Дата сессии		
FasilitationReason	Причина сессии		
FasilitationCondition	Состояние сессии		
EmplID	Уникальный идентификатор сотрудника инициатора сессии фасилитации		FK

Таблица 5 – Описание атрибутов сущности «vocation»

Имя атрибута	Назначение	Первичный ключ	Внешний ключ
VocationID	Уникальный идентификатор отпуска	РК	

Имя атрибута	Назначение	Первичный ключ	Внешний ключ
TypeVocation	Тип отпуска (описание причины)		
StartVocation	Дата начала отпуска		
EndVocation	Дата конца отпуска		
EmplID	Уникальный идентификатор сотрудника		FK

Таблица 6 – Описание атрибутов сущности «hr»

Имя атрибута	Назначение	Первичный ключ	Внешний ключ
HRID	Уникальный идентификатор роли сотрудника отдела кадров	PK	
HRDescription	Подробная информация о роли сотрудника отдела кадров		
JobTitle	Позиция в компании		
EmplID	Уникальный идентификатор сотрудника		FK

Таблица 7 – Описание атрибутов сущности «teamleader»

Имя атрибута	Назначение	Первичный ключ	Внешний ключ
TLID	Уникальный идентификатор роли лидера команды	PK	
TLDescription	Подробная информация о роли лидера команды		
JobTitle	Позиция в компании		
EmplID	Уникальный идентификатор сотрудника		FK

Таблица 8 – Описание атрибутов сущности «softwaredeveloper»

Имя атрибута	Назначение	Первичный ключ	Внешний ключ
SWDID	Уникальный идентификатор роли сотрудника команды	РК	
SWDDescription	Подробная информация о роли сотрудника команды		
JobTitle	Позиция в компании		
EmplID	Уникальный идентификатор сотрудника		FK

Таблица 9 – Описание атрибутов сущности «team»

Имя атрибута	Назначение	Первичный ключ	Внешний ключ
TeamID	Уникальный идентификатор команды	РК	
Name	Наименование команды		
StartDate	Начало работы команды		
EndDate	Окончание работы команды		
TLID	Идентификатор лидера команды		FK
ProjectID	Идентификатор проекта		FK
EmlCount	Количество сотрудников в команде		

Таблица 10 – Описание атрибутов сущности «project»

Имя атрибута	Назначение	Первичный ключ	Внешний ключ
PrjID	Уникальный идентификатор проекта	РК	
Name	Наименование проекта		
Description	Описание проекта		
Level	Уровень проекта		FK
Category	Категория проекта		
StartDate	Начало работы проекта		
EndDate	Окончание работы проекта		

Таблица 11 – Описание атрибутов сущности «swd_to_team»

Имя атрибута	Назначение	Первичный ключ	Внешний ключ
ID: INTEGER	Уникальный идентификатор записи	РК	
TeamID	Уникальный идентификатор команды		FK
SWDID	Уникальный идентификатор разработчика		FK

Таблица 12 – Описание атрибутов сущности «project_weekday»

Имя атрибута	Назначение	Первичный ключ	Внешний ключ
PrjID	Уникальный идентификатор проекта	РК	FK
WeekDayID	Уникальный идентификатор недели		FK

Имя атрибута	Назначение	Первичный ключ	Внешний ключ
Hourse	Количество часов потраченных на проект в неделе		

Таблица 13 – Описание атрибутов сущности «weekday»

Имя атрибута	Назначение	Первичный ключ	Внешний ключ
WkDayID	Уникальный идентификатор недели	РК	
Name	Наименование проекта		

Центральной таблицей в базе данных является таблица «employee». В данной таблице хранится информация обо всех сотрудникам организации занятых в проектной работе.

С таблицей «employee» связаны таблицы кадровой поддержки сотрудников:

– таблица «dismissal». Отвечает за информацию об увольнении сотрудника с фиксацией причины, почему это произошло;

– таблица «vocation». Отвечает за информацию об отпусках конкретного сотрудника и их причине;

– таблица «fasilitation». Отвечает за фиксацию сессий фасилитации и причины их обсуждений.

Таблица сотрудника связана с таблицами конкретных рабочих ролей: разработчик - таблица «softwaredeveloper», лидер команды – таблица «teamleader», сотрудник отдела кадров – таблица «hr», которые связывают сотрудников с их непосредственными обязанностями и конкретными активностями в рамках рабочего процесса.

Таблица команды «team» связывает разработчиков «softwaredeveloper» лидеров команды «teamleader» с проектами (таблица «project»).

При этом разработчики связаны с таблицей команды промежуточной таблицей «swd_to_team», так как один и тот же разработчик запишется более чем в одну команду, и в одной команде может быть множество разработчиков.

Таблицы «project» и «weekday» связаны между собой таблицей посредником project_weekday в связи с тем что каждый будний день команда работает более чем с одним проектом.

Таким образом, была разработана модель данных для СУБД MySQL.

4.2 Реализация программного приложения

Программное обеспечение, реализованное в дипломном проекте, имеет архитектуру на базе паттерна «Модель Вид Контроллер».

Обработка запросов реализованном приложении производится с помощью PHP роутеров.

Роутером в приложении являются компоненты, которые обеспечивают маршрутизацию HTTP-запросов к определенным обработчикам или контроллерам приложения. В реализованном приложении роутер - это набор правил, которые сопоставляют HTTP-запросы с конкретными обработчиками или контроллерами запросов, обеспечивая структурированную и организованную навигацию внутри приложения.

Ключевая особенность реализованного роутера:

- поддерживает методы HTTP протокола, такие как GET, POST, PUT, DELETE;

- позволяет создавать динамические шаблоны маршрутов с использованием регулярных выражений;

- поддерживает обработку ошибок 404 до и после прохождения маршрута.

Базовый PHP-маршрутизатор реализован с использованием файла index.php, который перенаправляет все запросы, используя файл .htaccess конфигурации. При этом пути маршрутизации включают определение маршрутов для

конкретных URL-адресов и связывание их с функциями обратного вызова или методами контроллера.

Реализованные PHP-роутеры выдают исключения, когда маршрут не найден или когда возникают проблемы с контроллером, который должен его обработать.

Ниже приведена часть реализованного роутера, который обрабатывает запросы о сотрудниках и командах.

```
php
<?php
require_once 'hrms/autoload.php';
use Phroute\Phroute\RouteCollector;
$router = new RouteCollector();
```

Первая часть роутера описывает коллекции запросов для всевозможной выборки информации о сотрудниках.

```
// описание роутеров для обработки данных о сотрудниках
$router->get('/employees', ['Controllers\EmployeeController',
'index']);
$router->get('/employees/{id:i}', ['Controllers\EmployeeController',
'show']);
$router->post('/employees', ['Controllers\EmployeeController',
'store']);
$router->put('/employees/{id:i}', ['Controllers\EmployeeController',
'update']);
$router->delete('/employees/{id:i}',
['Controllers\EmployeeController', 'destroy']);
```

Вторая часть роутера описывает коллекции запросов для всевозможной выборки информации о командах.

```
$router->get('/teams', ['Controllers\TeamsController', 'index']);
$router->get('/teams/{id:i}', ['Controllers\ TeamsController',
'show']);
$router->post('/teams', ['Controllers\ TeamsController', 'store']);
$router->put('/teams/{id:i}', ['Controllers\ TeamsController',
'update']);
```

```

$router->delete('/Teams/{id:i}', ['Controllers\TeamsController',
'destroy']);
// процесс обработки URL роутерами и передача полученных данных
контроллеру
$dispatcher = new Phroute\Phroute\Dispatcher($router->getData());
$response=$dispatcher->dispatch($_SERVER['REQUEST_METHOD'],
parse_url($_SERVER['REQUEST_URI'], PHP_URL_PATH));
// вывод результатов обработки роутеров
echo $response;

```

В представленном коде Phroute – это класс для определения маршрутов для приложения. Маршруты определены для двух сущностей: сотрудников и команд.

Маршруты сотрудников:

- GET/employees: - отображает список всех сотрудников;
- GET/employees/{id}: отображает сведения о конкретном сотруднике с заданными id;
- POST /employees: создает нового сотрудника;
- PUT /employees/{id}: обновляет существующего сотрудника с указанными id;
- DELETE /employees/{id}: удаляет сотрудника с указанным id.

Маршруты команд:

- GET/teams: Отображает список всех команд;
- GET/teams/{id}: Отображает сведения о конкретном отделе с заданными id;
- POST /teams: Создает новый отдел;
- PUT /teams/{id}: Обновляет существующий отдел с указанными id;
- DELETE /teams/{id}: Удаляет отдел с указанным id.

Для обработки данных маршрутов требуется реализовать класс обработки алгоритмов запросов – контроллер. Каждый тип данных и маршруты для работы с ними будут связаны с контроллером. Приведем часть контроллера для обработки данных о сотрудниках.

Вначале требуется подключить в контроллере модели, отвечающие за обработку данных сотрудников, команды и проекта.

```
namespace App\Controllers;
use App\Models\Employee;
use App\Models\Team;
use App\Models\Project;
```

Далее требуется реализовать конкретный класс контроллер `class EmployeeController extends Controller.`

В классе `EmployeeController` для работы с данными о командах, сотрудниках и проектах создаются специальные переменные:

```
private $teamRepository;
private $employeeRepository;
private $projectRepository;
```

В конструкторе класса `EmployeeController` определяются начальные значения объектов отвечающие за обработку информации о команда, сотрудниках и проектах:

```
public function __construct(TeamRepository $teamRepository,
EmployeeRepository $employeeRepository, ProjectRepository
$projectRepository) {
    $this->teamRepository = $teamRepository;
    $this->employeeRepository = $employeeRepository;
    $this->projectRepository = $projectRepository;
}
```

Далее по порядку реализуются функции класса `EmployeeController`, которые нужны для формирования, вывода списка сотрудников, создания нового сотрудника, изменения информации о сотруднике, удаление сотрудника.

Метод `public function index()` формирует список всех сотрудников.

```
public function index() {
    $employees = $this->employeeRepository->findAll();
    return view('employees.list', ['employees' => $employees]);}
```

Метод `public function show($id)` показывает сотрудников с определенным номером ID.

```

public function show($id) {
    $employee = $this->employeeRepository->findById($id);
    return view('employees.show', ['employee' => $employee]);
}

```

Метод `public function create()` необходим для создания записи о сотруднике и формировании информации о нем, для записи в таблицу базы данных.

```

public function create() {
    if (isset($_POST['submit'])) {
        $employee = new Employee();
        $employee->firstName = $_POST['firstName'];
        $employee->lastName = $_POST['lastName'];
        $employee-> jobtitle = $_POST['jobtitle'];
        $employee-> isactive = $_POST['isactive'];
        $employee-> secondname = $_POST['secondname'];
        $department = $this->teamRepository-
>findById($_POST['teamId']);
        $this->employeeRepository->save($employee);
        header('Location: /employees');
        exit;
    }
}

```

Для обновления информации о созданном сотруднике необходим метод `public function update($id)`.

```

public function update($id) {
    $employee = $this->employeeRepository->findById($id);
    if (isset($_POST['submit'])) {
        $employee = new Employee();
        $employee->firstName = $_POST['firstName'];
        $employee->lastName = $_POST['lastName'];
        $employee-> jobtitle = $_POST['jobtitle'];
        $employee-> isactive = $_POST['isactive'];
        $employee-> secondname = $_POST['secondname'];
        $department = $this->departmentRepository-
>findById($_POST['departmentId']);
        $this->employeeRepository->save($employee);
    }
}

```

```

header('Location: /employees');
exit;
}

$departments = $this->departmentRepository->findAll();
return view('employees.update', ['employee' => $employee, 'team' =>
$team, 'project' => $project]);
}

```

Метод `public function delete($id)` необходим для удаления записи о сотруднике и обновлении данного факта в базе данных.

```

public function delete($id) {
    $employee = $this->employeeRepository->findById($id);
    $this->employeeRepository->delete($employee);
    header('Location: /employees');
    exit;
}
}

```

Структура приложения на основании кода представленного выше выглядит следующим образом (рисунок 10).

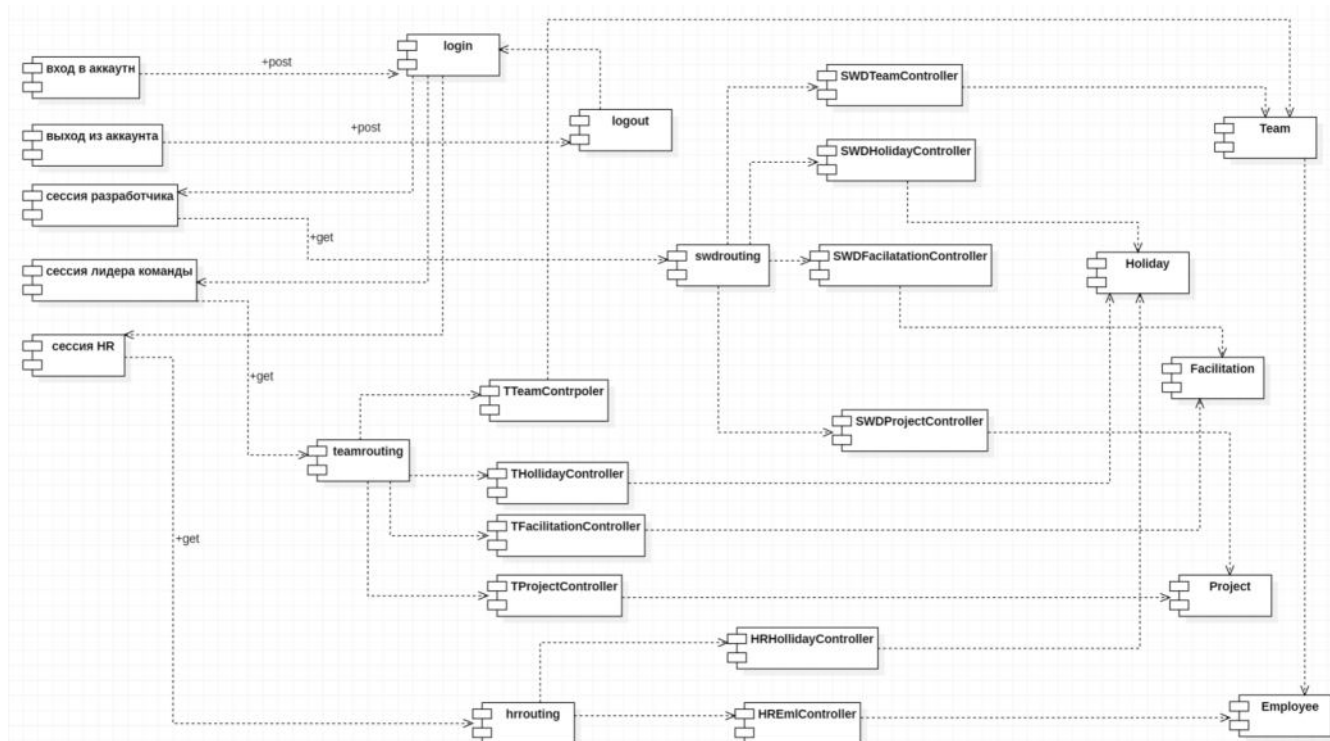


Рисунок 10 – Диаграмма компонентов приложения.

Запросы: вход в аккаунт, выход из аккаунта, рабочие сессии соответствующих ролей вызывают необходимый маршрут URL, который обрабатывается соответствующим маршрутизатором. Маршрутизатор формирует команду для соответствующего контроллера, который обрабатывает запрос и если необходимо запрашивает модель (Team, Holiday? Facilitation, Project) для CRUD операции с базой данных.

4.3 Реализация интерфейса программы

Презентационная часть WEB-приложения состоит из следующих страниц, которые отвечают за работу приложения в одном из режимов: разработчик, лидер команды, сотрудник отдела кадров.

На рисунках 11 – 18 показана работа в режиме сотрудника команды (разработчик).

					ПГУ 1.090501.05.001 ПЗ	Лист
						47
Изм.	Лист	№ докум.	Подпись	Дата		

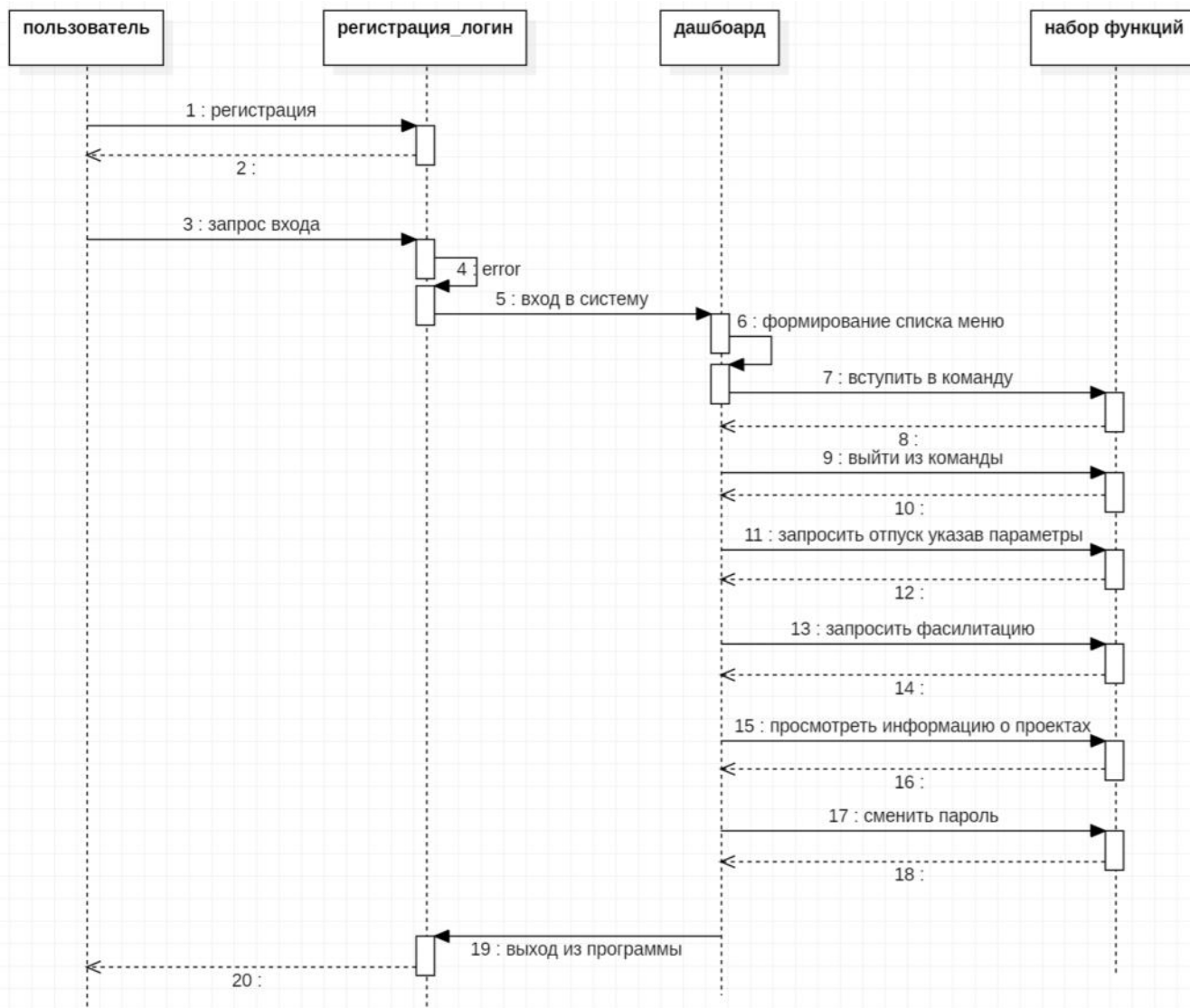


Рисунок 11 – Диаграмма последовательности. Работа в режиме разработчика.

Для входа в систему пользователю необходимо иметь аккаунт и e-mail с паролем в систему (рисунок 12).

Рисунок 12 – Вход сотрудника (разработчик)

После успешного входа появляется главная страница приложения (рисунок 13.)

Рисунок 13 – Главная страница (разработчик)

С помощью инструментов на главной странице пользователь может пройти по ссылке и сформировать маршрут для: работы с данными о команде, с отпуском

для своих нужд, вызвать сессию фасилитации или посмотреть проекты, в которых пользователь участвует.

Рисунок 14 – Информационная панель команды (разработчик)

Разработчику предоставляется возможность войти в существующие команды и выйти из них, помимо цифрового действия данный процесс согласуется с лидером команды и HR с помощью бюрократических механизмов – служебная записка и приказ.

Рисунок 15 – Запрос на предоставление отпуска от разработчика

Запрос на отпуск дает возможность сотруднику обозначить его потребности без личного присутствия в офисе (рисунок 15).

Запрос на фасилитацию необходим для решения спорных и острых вопросов в рамках команды или отдела с привлечением специалистов по фасилитации и возможно коучей (рисунок 16).

Рисунок 16 – Запрос на фасилитацию от разработчика

Команда	Инициатор ФИО	Инициатор ID
Команда А	Иванов Иван	1234

Рисунок 17 – Фиксация запроса на фасилитацию от разработчика

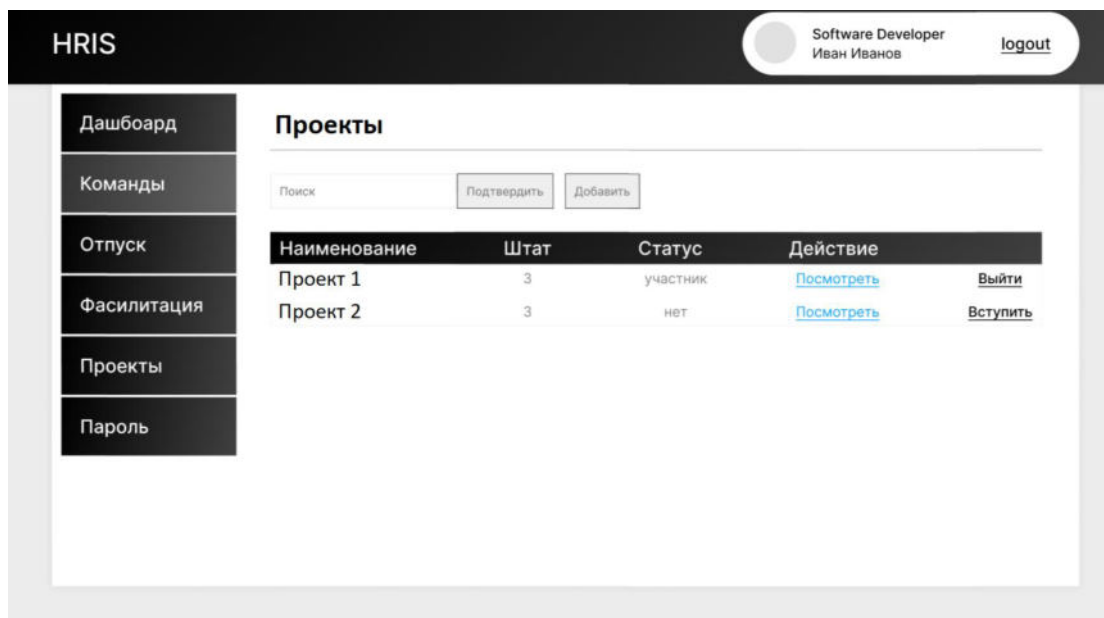


Рисунок 18 – Информационная панель проектов разработчика

Разработчику предоставляется возможность просматривать информацию о проекте и его участниках.

На рисунках 19 – 25 показана работа в режиме лидера команды (teamlider).

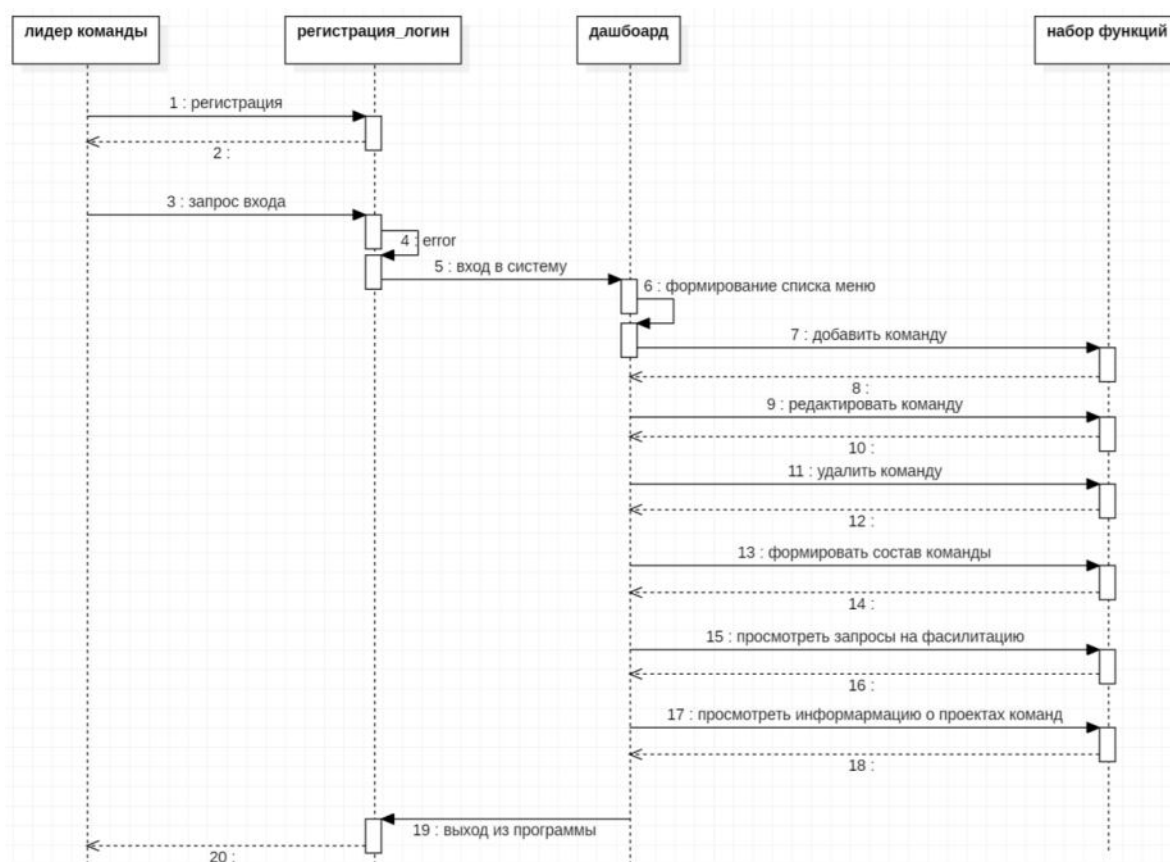


Рисунок 19 – Диаграмма последовательности. Работа в режиме teamleader.

Вход в систему с ролью teamlider показан на рисунке 20.

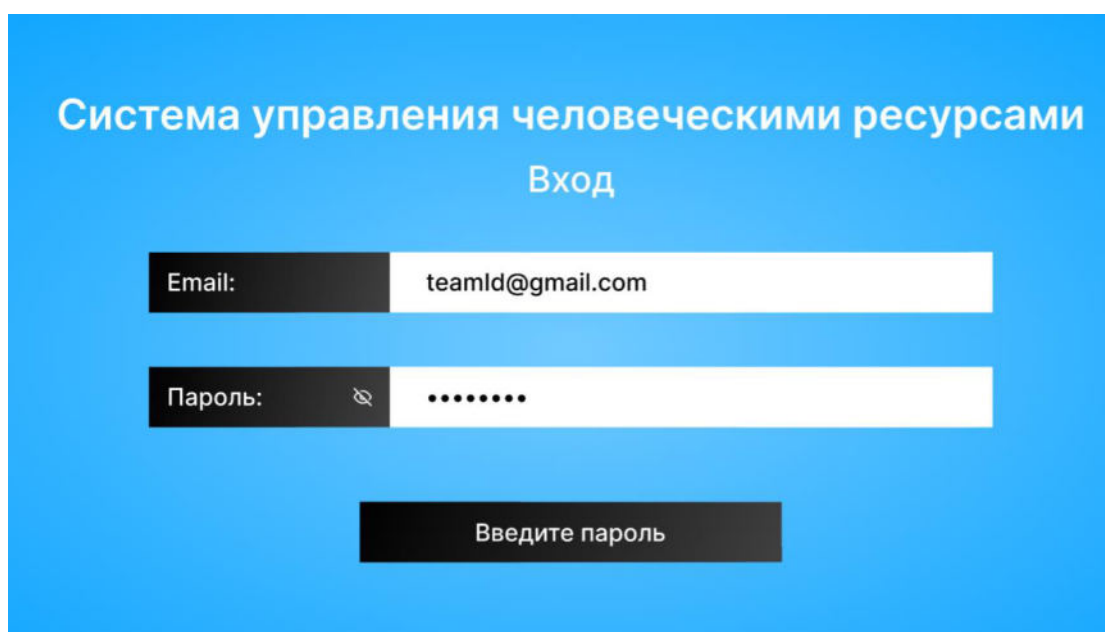
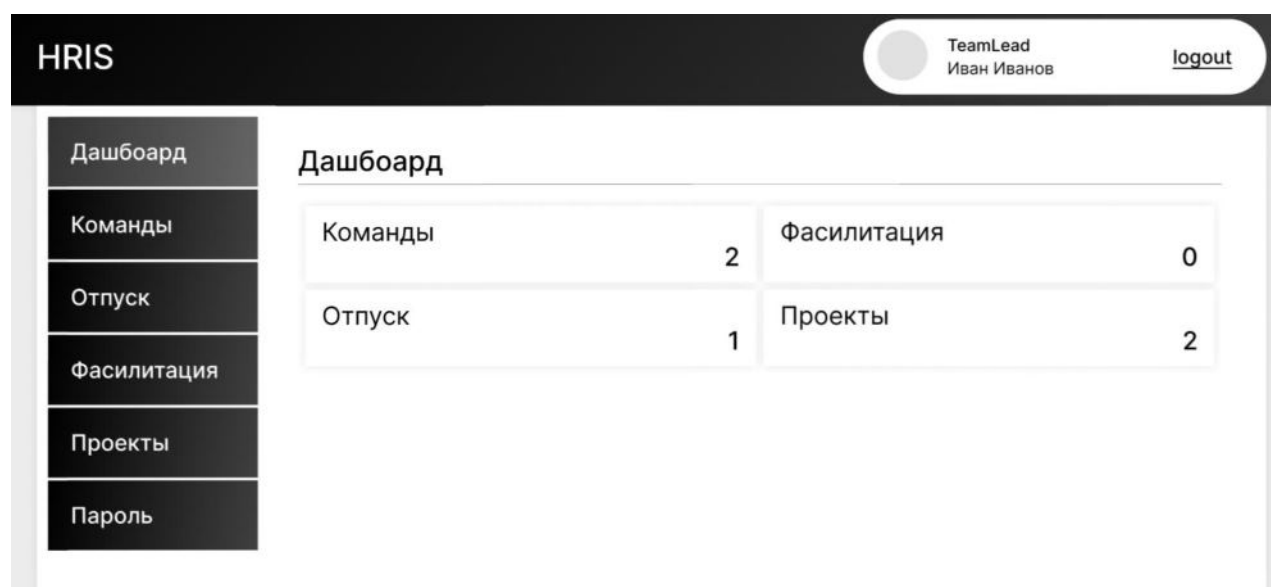


Рисунок 20 – Вход лидера команды (teamlider).

На основной странице (рисунок 21) представлен инструментарий управления командами и проектами для лидера команды (teamlider).



Дашбоард	
Команды	2
Фасилитация	0
Отпуск	1
Проекты	2

Рисунок 21 – Главная страница лидера команды (teamlider).

Выбрав пункт меню «Команды», пользователь видит список сформированных команд и имеет полномочия на просмотр и изменения состава команд. Также есть возможность аннулировать команду (рисунок 22).

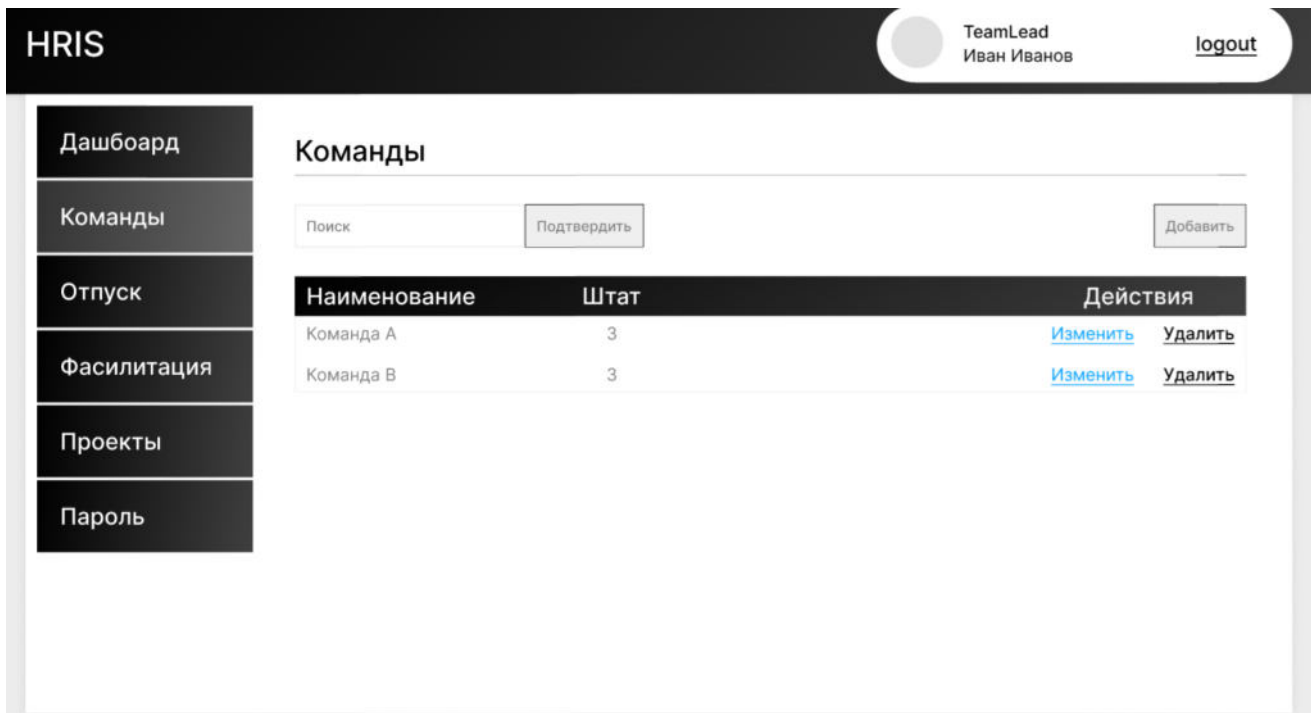


Рисунок 22 – Страница управления командами лидера команды (teamlider).

Кнопка «Добавить» (рисунок 23) дает возможность сформировать новую команду.

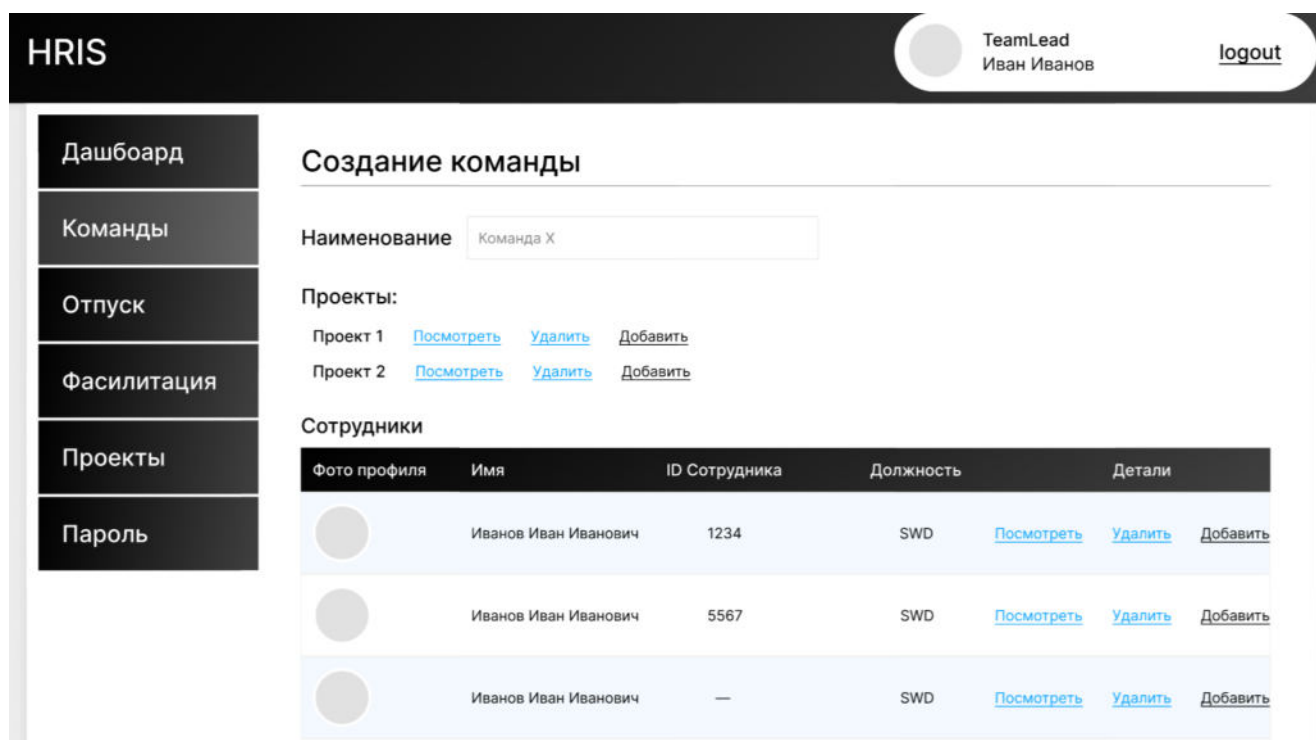


Рисунок 23 – Создание команды лидера команды (teamlider).

В любой доступной команде лидер может произвести кадровые перестановки (рисунок 24).

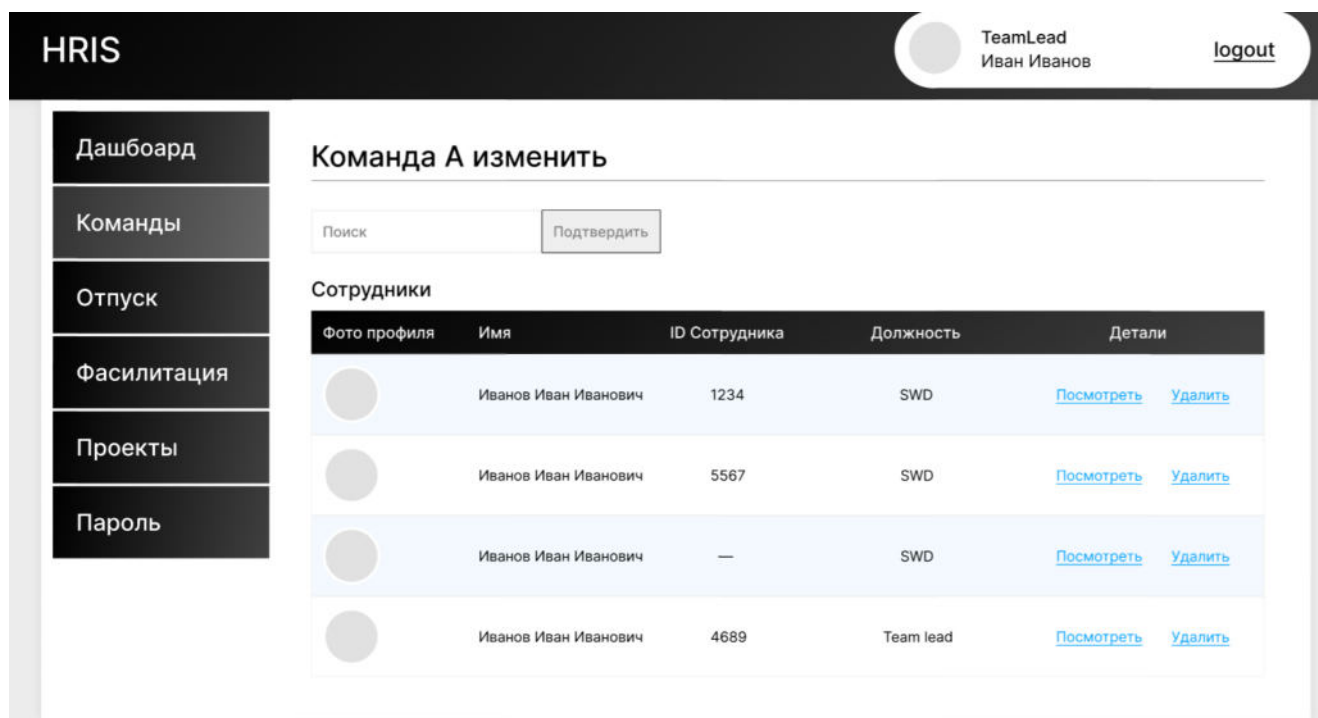


Рисунок 24 – Форматирование команды лидера команды (teamlider).

Также имеется возможность посмотреть запрос на фасилитацию и принять административные меры для решения возникших организационных вопросов (рисунок 25).

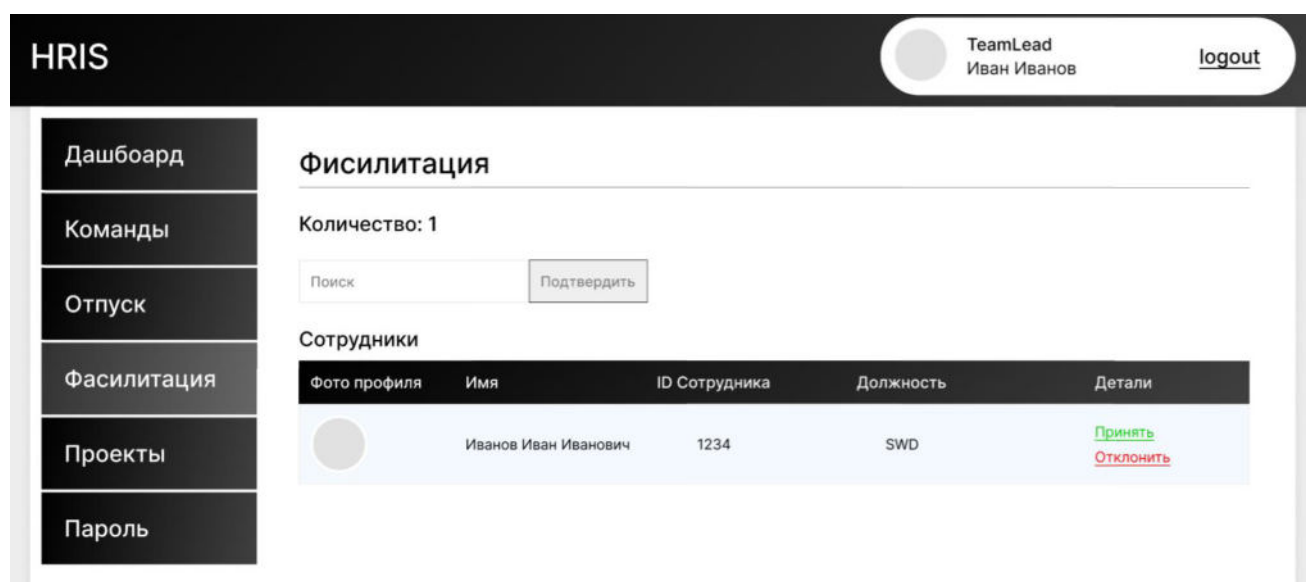


Рисунок 25 – Решение по фасилитации лидера команды (teamlider).

На рисунках 26 – 29 показана работа в режиме сотрудника отдела кадров (human resouse - HR).

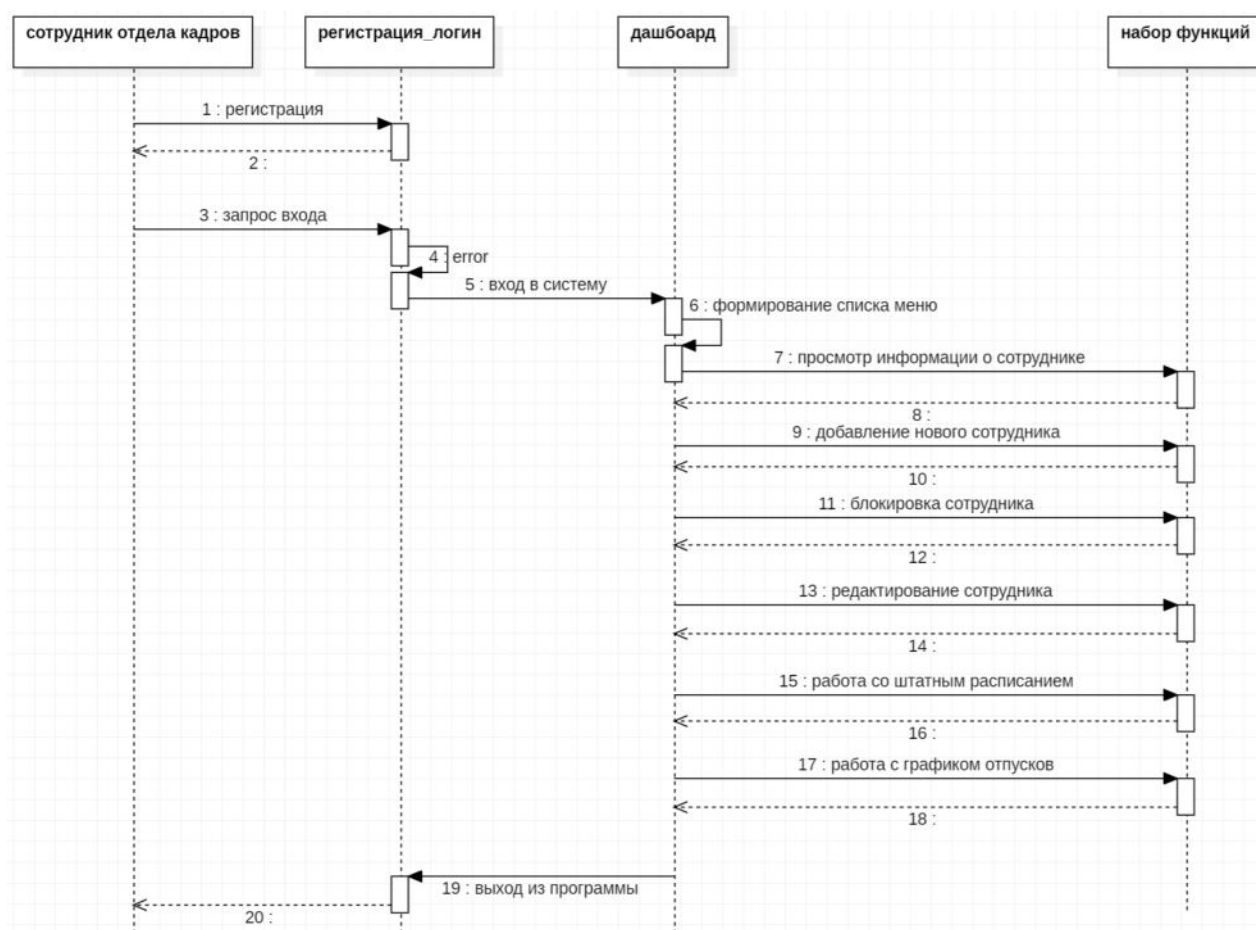


Рисунок 26 – Диаграмма последовательности.

Работа в режиме сотрудника отдела кадров.

Система управления человеческими ресурсами

Вход

Email:

Пароль:

✖

Введите пароль

Рисунок 27 – Вход HR.

После аутентификации пользователь HR попадает на страницу с инструментами (рисунок 28). На которой ему предоставляется возможность работать с информацией о сотрудниках.

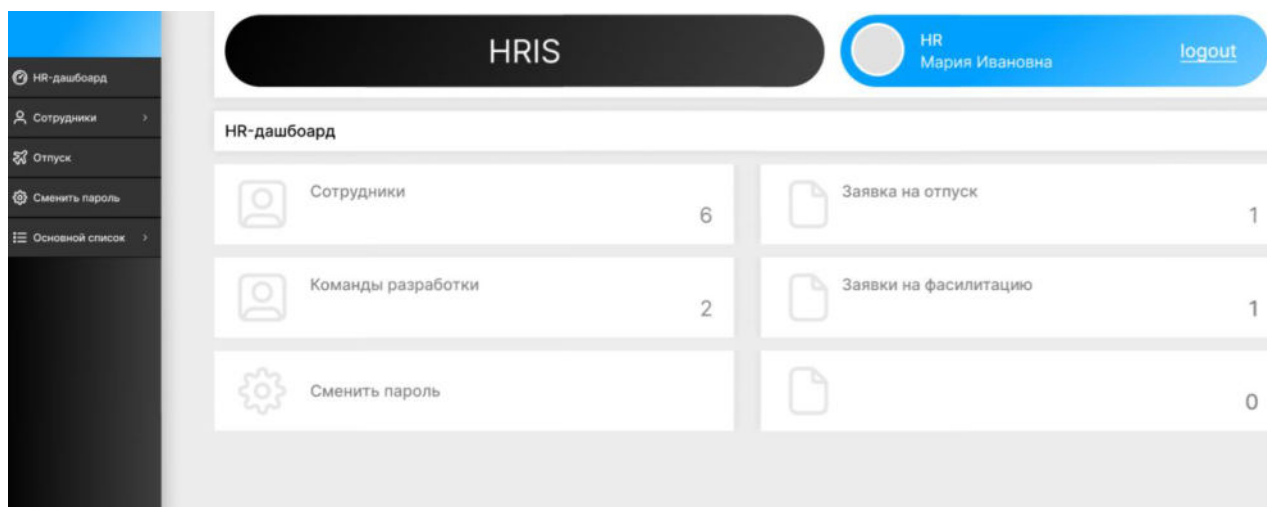


Рисунок 28 – Основная панель инструментов HR.

Перейдя к списочному составу сотрудников (рисунок 29).

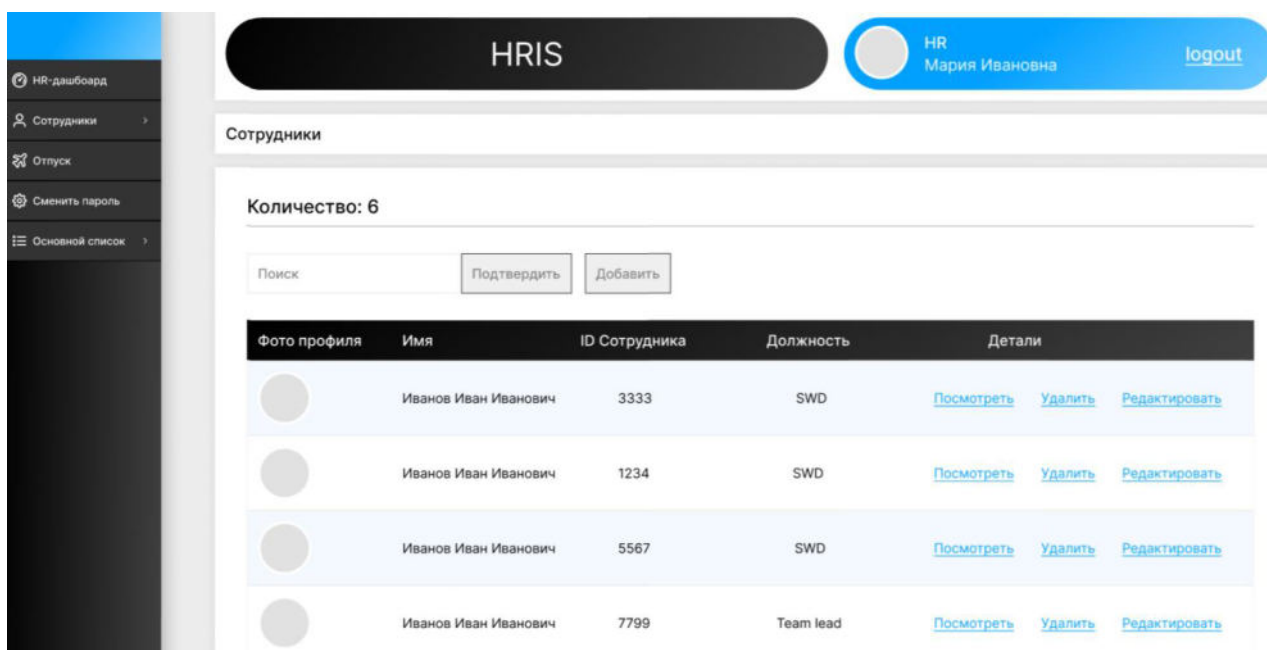


Рисунок 29 – Панель работы с сотрудниками HR.

Сотрудник отдела кадров имеет возможность перейти к карточке конкретного сотрудника и произвести изменения в данных, вплоть до блокировки сотрудника (рисунок 30).

HR-дашборд

Сотрудники

Отпуск

Сменить пароль

Основной список

HRIS

HR Мария Ивановна

logout

Сотрудники > Редактирование

Фотография профиля

edit

Выбрать файл

Файл не выбран

ID работника

1234

Имя

Иван

Фамилия

Иванов

Отчество

Иванович

Номер телефона

+7 999 318 5281

Email

ivan@mail.ru

Пол

edit

Мужчина

Дата рождения

01.12.2000

Супруг

Женат

Позиция

Software Developer (SWD)

Дата присоединения

20.06.2019

Команда

Команда А

Пароль

Пароль

Проекты: 2

read only

Проект 1

Проект 2

Фасилитация

Да

Отпуск

14.09.2024

Подтвердить

Сбросить

Заблокировать

Рисунок 30 – Панель работы с конкретным сотрудником HR.

Сотрудник отдела кадров имеет возможность управления отпусками. Ему доступны три режима: просмотр подтверждение или отклонение запроса. Любой статус может быть изменен по требованию сотрудника или руководства.

HR-дашборд

Сотрудники

Отпуск

Сменить пароль

Основной список

HRIS

HR Мария Ивановна

logout

Отпуск

Запрос отпуска

ID	Имя	Тип отпуска	Начало	Конец	Действие
1234	Иванов Иван	Обычный отпуск	14.09.2024	02.10.2024	Подтвердить Отклонить

Подтвержденный отпуск

ID	Имя	Тип отпуска	Начало	Конец	Действие
----	-----	-------------	--------	-------	----------

Отклоненный отпуск

ID	Имя	Тип отпуска	Начало	Конец	Действие
----	-----	-------------	--------	-------	----------

Рисунок 31 – Панель обработки запросов на отпуск HR.

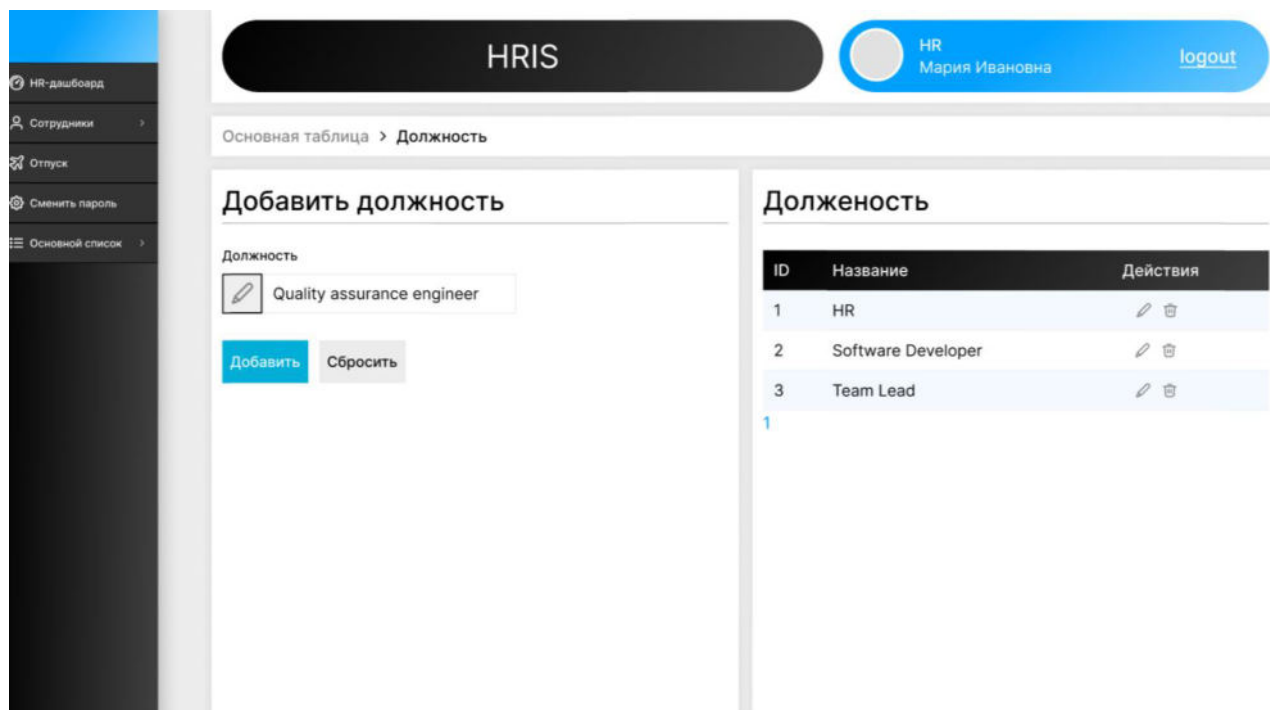


Рисунок 32 – Панель работы со штатным расписанием HR.

Также в задачу сотрудника отдела кадров входит формирование штатного расписания для команд (рисунок 32).

5 Оценка затрат на разработку и внедрение программного приложения контроля выполнения задач

5.1 Организация производства разработки программного приложения

Разработка программного приложения осуществлена на базе собственной площади общей площадью 30 кв.м. Содержание помещения в месяц составляет 15000 рублей.

Для разработки программного продукта приобретено оборудование (вычислительная техника), лицензионное программное обеспечение, дополнительные материалы.

Данные по потребности в оборудовании и лицензионном программном обеспечении представлены соответственно в таблицах 13 и 14.

Таблица 13 – Потребность в оборудовании

Оборудование	Источник	Количество	Ориентировочная цена за единицу, рублей	Потребляемая мощность, Ватт
Ноутбук Huawei MateBook D 14 NbD-WDI9, 14"	ООО «Ситилинк»	3	39990	136

Таблица 14 – Потребность в лицензионном программном обеспечении

Лицензионное программное обеспечение	Источник	Цена, рублей	Срок действия лицензии, дней
Microsoft Windows 10	Комплект «Ноутбук Huawei, 14"»	-	бессрочная

Лицензионное программное обеспечение	Источник	Цена, рублей	Срок действия лицензии, дней
VS Code	Интернет-магазин VS Code.	-	бессрочная
Антивирус Kaspersky Plus 2023	Интернет-магазин Kaspersky	1990	365
СУБД MySQL	Интернет-магазин MySQL	-	бессрочная

Перечень дополнительных материалов, источники их приобретения, объем, и ориентировочные цены приведены в таблице 15.

Таблица 15 – Потребность в дополнительных материалах

Материал	Источник	Количество, штук	Ориентировочная цена за единицу, рублей
Стол компьютерный ДИЗАЙН ФАБРИКА	ООО «Леруа Мерлен	3	2000
Рабочая лампа	ООО «Леруа Мерлен восток»	3	1150
Стул Leset Норман	ООО «Леруа Мерлен восток»	3	5200
Канцтовары	Гипермаркет «Канцлер»	3	1000

Разработка программного приложения контроля выполнения задач осуществлялась командой разработчиков, представленных в таблице 16.

Таблица 16 – Состав коллектива разработчиков

Специалист	Количество, человек	Описание занятости
Руководитель проекта	1	<ul style="list-style-type: none"> - управление проектом - набор команды исполнителей - управление ресурсами проекта - контроль соблюдения требований заказчика - приемка и внедрение проекта - анализ предметной области
Программист	1	<ul style="list-style-type: none"> - создание базы данных - написание, отладка и первичное тестирование программного кода - разработка и утверждение программной документации - приемка и внедрение программного кода
Тестировщик	1	<ul style="list-style-type: none"> - тестирование программного приложения

Данные по основной заработной плате разработчиков программного приложения и их количественному составу приведены в таблице 17.

Таблица 17 – Сведения о заработной плате

Специалист	Количество, человек	Сумма заработной платы, рублей
Руководитель проекта	1	50000
Программист	1	50000
Тестировщик	1	25000

5.2 Расчет себестоимости программного обеспечения для программного приложения контроля выполнения задач

Себестоимость разработки программного приложения включает следующие расходы:

- на приобретение оборудования;
- на приобретение лицензионного программного обеспечения;
- затраты на дополнительные материалы;
- затраты на заработную плату;
- затраты на обслуживание программы.

Затраты на оборудование в период разработки составят 119970 рублей и представлены в таблице 13.

Затраты на программное обеспечение в период разработки составят 1990 рублей и представлены в таблице 14.

Затраты на дополнительные материалы в период разработки составят 28050 рублей и представлены в таблице 15.

Затраты на заработную плату включают затраты на основную заработную плату разработчиков программного приложения и отчисления на страховые взносы.

С учетом выходных дней при 5-дневной рабочей неделе количество рабочих дней в месяце в среднем составляет 21 день. На разработку всего проекта необходимо 3 месяца.

Таким образом, затраты на заработную плату разработчиков составят 125000 рублей в месяц, 375000 рублей за весь период создания проекта.

Отчисления на страховые взносы – обязательные отчисления по нормам, установленным законодательством государственного социального страхования, в Пенсионный фонд РФ, фонд обязательного медицинского страхования, фонд социального страхования РФ от затрат на оплату труда работников, включаемых в себестоимость продукции (программного обеспечения), по элементу «Затраты на

					ПГУ 1.090501.05.001 ПЗ	Лист
						63
Изм.	Лист	№ докум.	Подпись	Дата		

оплату труда» (кроме тех видов оплаты, на которые страховые взносы не начисляются).

Отчисления на страховые взносы производится согласно «Налоговому кодексу Российской Федерации» от 14.07.2022 № 239-ФЗ.[11]

С января 2023 года введена единая базовая ставка по страховым взносам. Действующие тарифы на страховые взносы работодателей установлены статьей 425 НК РФ и составляют 30%.

Так как компания осуществляет деятельность в области ИТ, то она может претендовать на льготную ставку (подп. 3 п. 1 ст. 427 НК РФ) равную 7,6%.

Таким образом, с выплат в пределах МРОТ (минимального размера оплаты труда) уплачиваю взносы – 30%, с суммы свыше МРОТ уплачиваются взносы – 7,6%. МРОТ на 2023 год составляет 16242 рубля.

Данные по суммам отчислений на страховые взносы приведены в таблице 18.

Таблица 18 – Отчисления на страховые взносы

Сотрудник	Заработная плата, рублей	Взнос по единой ставке, рублей	Сумма, подлежащая пониженной ставке по взносу, рублей	Взнос по пониженной ставке, рублей	Общая сумма взноса с сотрудника за месяц, рублей
Руководитель проекта	50000	4872,6	33758	2565,6	7438,2
Программист	50000	4872,6	33758	2565,6	7438,2
Тестировщик	25000	4872,6	8758	665,6	5538,2

Затраты на заработную плату и отчисления на страховые взносы представлены в таблице 19.

Таблица 19 – Затраты на заработную плату

Сотрудник	Заработная плата, рублей	Общая сумма отчислений, рублей	Всего за 1 месяц, рублей	Всего за 3 месяца, рублей
Руководитель проекта	50000	7438,2	57438,2	172314,6
Программист	50000	7438,2	57438,2	172314,6
Тестирующий	25000	5538,2	30538,2	91614,6
Итого:	125000	20416,6	145414,6	436243,8

Затраты на заработную плату команды проекта по созданию программного приложения составят 436243,8 рублей из которых общая сумма отчислений на страховые взносы разработчиков составит 20416,6 рублей, а заработная плата составит 415827,2 рублей.

Затраты на обслуживание включают следующие статьи:

- амортизация оборудования, составляющая 20% от стоимости оборудования;
- обслуживание оборудования, составляющее 15% от амортизационных отчислений;
- затраты на электроэнергию, включающие затраты на электроэнергию персональных компьютеров, оргтехники и бытовых приборов;
- расходы на содержание помещения;
- накладные расходы.

Амортизация оборудования составляет 20% от стоимости оборудования. Было приобретено 3 ноутбука по 39990 рублей, Таким образом затраты на амортизацию оборудования в месяц составят:

$$\frac{(3 * 39990) * 0,2}{12} = 1999,5 \text{ руб}$$

Обслуживание оборудования составляет 15% от амортизационных отчислений.

Таким образом затраты на обслуживание оборудования составят:

$$1999,5 * 0,15 = 299,93 \text{ руб.}$$

Предположительно в день по 8 часов будут работать 3 ноутбука:

$$136 * 8 * 63 * 3 = 247 \text{ кВт.}$$

Таким образом, за 63 дня разработки программного приложения будет использовано 247 кВт электроэнергии на питание ноутбуков, а также 0,5 кВт на прочие приборы.

$$247 + 0,5 = 247,5 \text{ кВт.}$$

Накладные расходы составляют 75% то заработной платы. Таким образом, затраты на накладные расходы составляют:

$$415827,2 * 0,75 = 311870,4 \text{ руб.}$$

Результат расчетов по статьям затрат на обслуживание приведены в таблице 20.

Таблица 20 – Затраты на обслуживание

Статья затрат	Показатель отчислений	Сумма отчислений в месяц, рублей	Итоговая сумма отчислений за период разработки, рублей
Амортизация оборудования	20%	1999,5	5998,5
Обслуживание оборудования	15%	299,9	899,8
Затраты на электроэнергию	2,97 на 1 кВт/ч	735,1	2205,3
Накладные расходы	75%	103956,8	311870,4

Статья затрат	Показатель отчислений	Сумма отчислений в месяц, рублей	Итоговая сумма отчислений за период разработки, рублей
Аренда офиса и расходы на его содержание	-	15000	45000
Итого, рублей		121991,3	365973,9

Себестоимость разработки определяется как сумма затрат, включающая все статьи затрат на реализацию проекта, представленное в таблице 21.

Таблица 21 – Общая сумма затрат

Статья затрат	Сумма отчислений, рублей
Приобретение оборудования	119970
Приобретение лицензионного программного обеспечения	1990
Затраты на заработную плату	436243,8
Статья затрат	Сумма отчислений, рублей
Затраты на дополнительные материалы	28050
Затраты на обслуживание	365973,9
Итого:	952227,7

Таким образом, сумма затрат на разработку проекта составит 952227,7 рублей и будет соответствовать себестоимости программного приложения контроля выполнения задач.

6 Общие принципы обеспечения безопасности на рабочем месте оператора программного приложения

6.1 Актуальность вопросов безопасности человека на рабочем месте

С развитием и большим внедрением IT технологий в жизнь человека, большое распространение получили ПЭВМ (персональные электронные вычислительные машины). Они стали применяться и для повышения производительности труда сотрудников в разных сферах. Именно поэтому актуальным вопросом является безопасность оператора ПЭВМ, а также сохранение его здоровья и работоспособности.

Использование ЭВМ связано с распознаванием печатного и рукописного текста, анализ изображений на экране, долгим нахождением в одной позе, и др. Это может пагубно сказаться на здоровье человека, если не придерживаться нормами использования ЭВМ.

Согласно ГОСТ 12.0.003-74[11] вредные производственные факторы подразделяются на 4 категории:

- физические, к ним относятся недостаточная освещенность рабочего места, повышенная яркость света, отклонения от нормы температуры воздуха, высокая влажность воздуха, повышенный уровень шума, и др.;
- психофизиологические, к ним относятся нервно-психические и физические перегрузки, перенапряжение ушей и глаз;
- биологические, к ним относятся микроорганизмы и макроорганизмы, которые могут вызвать различные заболевания;
- химические, к ним относятся превышения в рабочей зоне аммиака, фенола, озона, оксида углерода и др.

6.2 Влияние ПЭВМ на физиологию человека

Работа за компьютером подразумевает длительное напряжение на зрительные органы и опорно-двигательный аппарат. Если не контролировать время пребывания за ЭВМ, это может привести к нарушению зрения, искривлению позвоночника, а также нарушению кровообращения в организме.

Нагрузка на зрительный аппарат может вызвать нарушения в работе ЦНС (центральной нервной системы). При длительной работе за компьютером снижается фокусировка и острота зрения, увеличивается чувствительность глаз.

Длительное нахождения человека в одной позе приводит к быстрому развитию утомляемости. При работе за компьютером стоит обратить внимание на эти вещи:

- высота рабочей зоны и стула;
- наличие опорной спинки и подлокотников;
- угол наклона головы при работе;
- угол наклона экрана компьютера;
- пространство и подставка для ног.

Большое влияние на здоровье человека оказывает и уровень шума в помещении. Согласно СанПиН 2.2.3670-20 "Санитарно-эпидемиологические требования к условиям труда", эквивалентный уровень шума не должен превышать 50 дБА. При длительном превышении уровня шума в помещении ухудшаются условия труда человека, а также оказывается вред здоровью. Происходит снижение концентрации внимания и повышается усталость, следовательно, снижается работоспособность сотрудника. Чтобы снизить уровень шума в помещении устанавливается звукоизоляция.

6.3 Техника безопасности при работе с персональным компьютером

Согласно типовой инструкции по охране труда при работе на ПК (персональном компьютере) ТОИ Р-45-084-01 [15], которая была утверждена

					<i>ПГУ 1.090501.05.001 ПЗ</i>	Лист
						69
Изм.	Лист	№ докум.	Подпись	Дата		

Приказом Минсвязи Российской Федерации №162 от 02.07.2001, деятельность на ЭВМ делится на 3 группы:

- группа А. Чтение информации с монитора подразумевает два перерыва по 15 минут, без учета обеденного перерыва;

- группа Б. Набор текста подразумевает перерывы по 10 минут раз в час;

- группа В. Творческий труд подразумевает перерывы по 15 минут раз в час.

Данное время рекомендуется уделять физической нагрузке, такой как зарядка или гимнастика для глаз.

Подводя итог, для сохранения здоровья сотрудников и ограничения от негативного влияния ПЭВМ необходимо использовать только технику, обладающую сертификатами качества, а также обеспечивать сотрудникам рациональный режим труда и перерывов. Важно заметить, что сертифицированная техника – это не полная гарантия безопасности условий труда, необходимо регулярно проводить проверку всей используемой техники.

					<i>ПГУ 1.090501.05.001 ПЗ</i>	Лист
						70
Изм.	Лист	№ докум.	Подпись	Дата		

Заключение

В ходе выполнения дипломного проекта была изучена предметная область, и определена актуальность разработки программного обеспечения. На основе анализа было сформулировано техническое задание для разработки программного приложения. Далее были выбраны средства разработки, благодаря которым проект был разработан.

Программное приложение реализовано в соответствии с техническим заданием.

В разделе экономики была произведена оценка затрат на разработку и внедрение приложения.

В разделе безопасность жизнедеятельности были проанализированы и описаны общие принципы обеспечения безопасности на рабочем месте оператора программного приложения контроля выполнения задач.

Задание на дипломное проектирование выполнено в полном объеме.

					<i>ПГУ 1.090501.05.001 ПЗ</i>	Лист
						71
Изм.	Лист	№ докум.	Подпись	Дата		

Список использованных источников

1. Statista – глобальная платформа данных и бизнес-аналитики с обширной коллекцией статистических данных, отчетов и аналитической информации [Электронный ресурс] – <https://www.statista.com/outlook/tmo/software/worldwide#revenue> (дата обращения 12.03.2024)
2. Официальный сайт Visual Studio Code [Электронный ресурс] – <https://code.visualstudio.com/> (дата обращения 01.03.2024).
3. Фримен Элизабет, Фриман Эрик – «Изучаем программирование на HTML5». – 2013. – 592с.
4. Макфарланд Д. С. – «Новая большая книга CSS». – 2016. – 720с.
5. Еременко А.В. – «Проектирование баз данных с помощью OpenModelSphere». – 2011. – 73с.
6. Карпова И.П. – «Базы данных. Курс лекций и материалы для практических заданий». – Учебное пособие. – 2013. – 240с.
7. СанПиН 2.2.2/2.4.1340-03. Библиографическая запись. Гигиенические требования к персональным электронно-вычислительным машинам и организации работы. Изд-во стандартов, 2003. – 41 с.
8. Трудовой кодекс Российской Федерации об утверждении перечней вредных и (или) опасных производственных факторов и работ № 302. – 2011.
9. Типовая инструкция по охране труда при работе на персональном компьютере. – 2001. – 3 с.

Приложение А

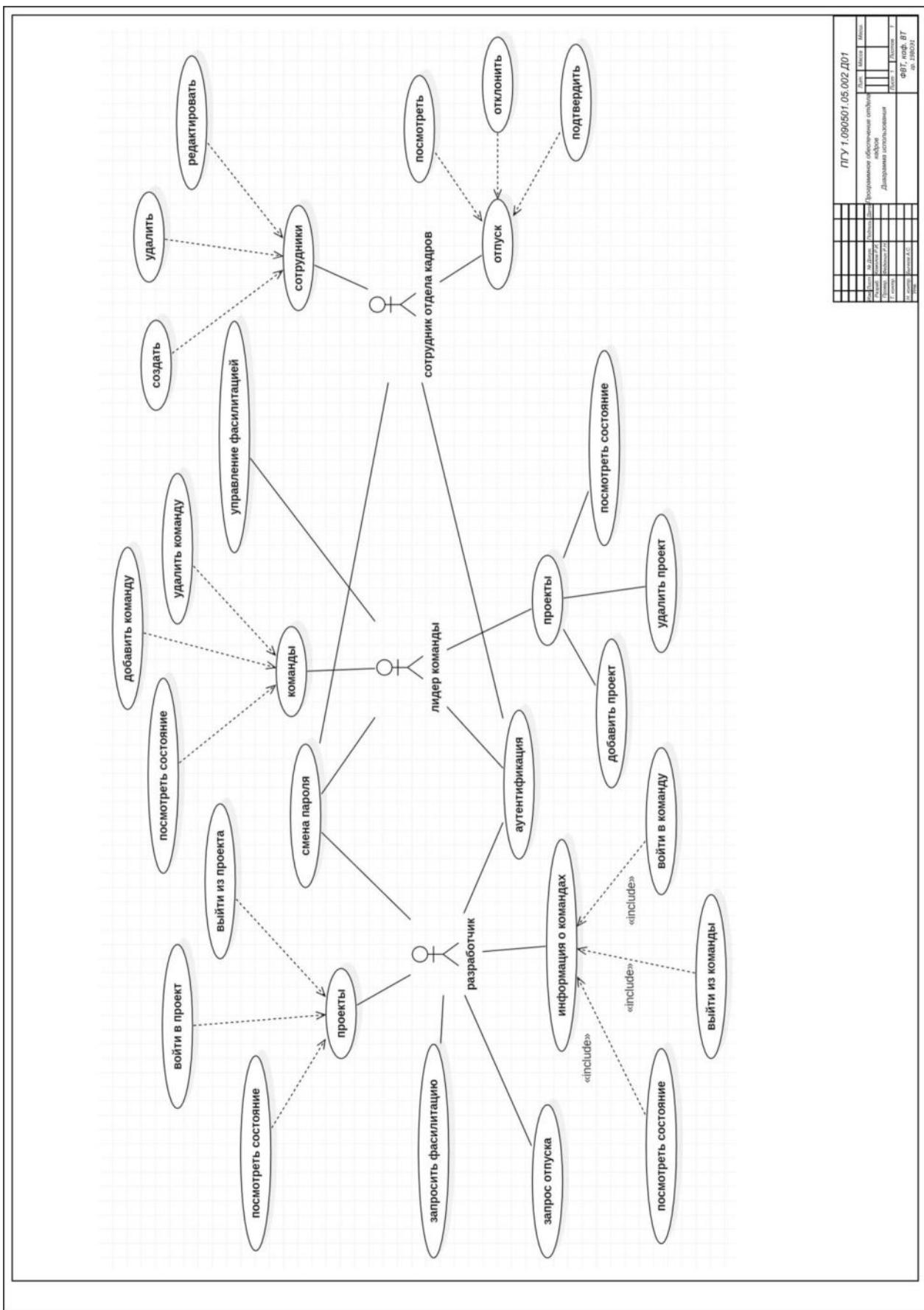
(обязательное)

Программное обеспечение отдела кадров

UML-диаграммы

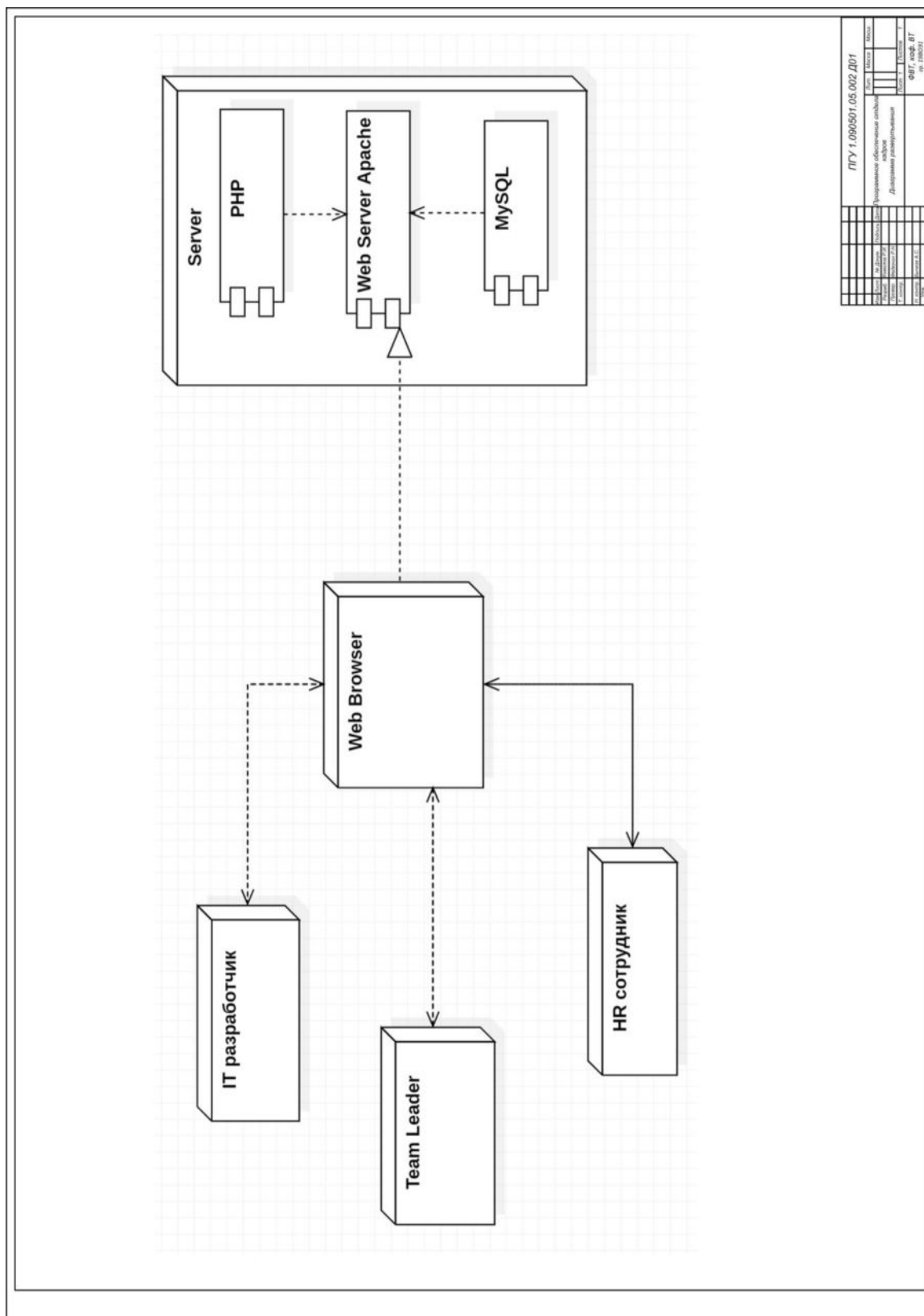
					<i>ПГУ 1.090501.05.001 ПЗ</i>	Лист
						73
Изм.	Лист	№ докум.	Подпись	Дата		

Приложение А.1 – Диаграмма вариантов использования

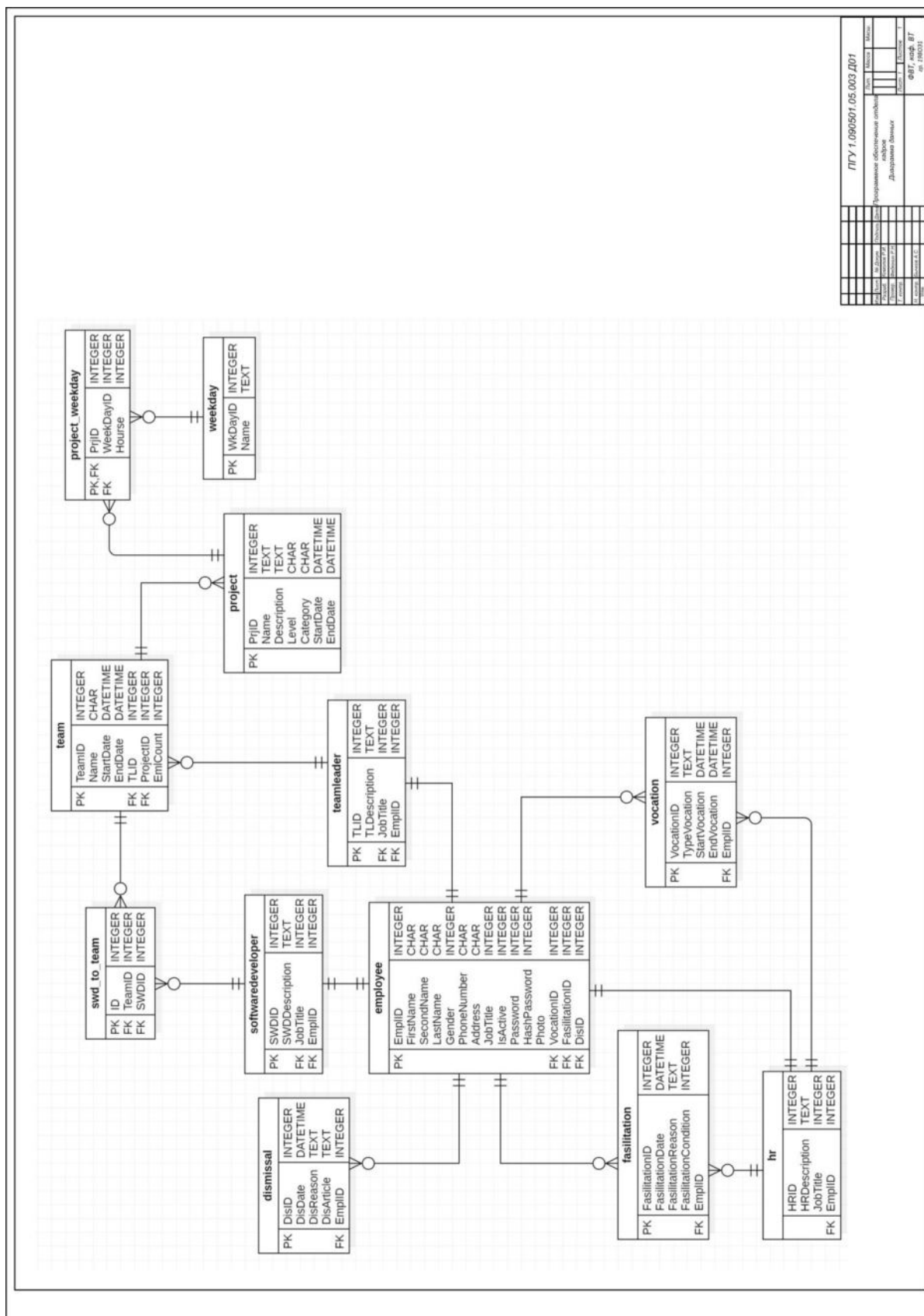


ПГУ 1.090501.05.002 Д01									
№ докум.	Исполн.	Дата	Проверен	Обсужден	Дата	Масштаб	Масштаб	Масштаб	Масштаб
1.090501.05.002	И.И.И.	2020.01.01	И.И.И.	И.И.И.	2020.01.01	1:1	1:1	1:1	1:1
Диаграмма использования									
ФГУ, №Ф. 87									
45-100002									

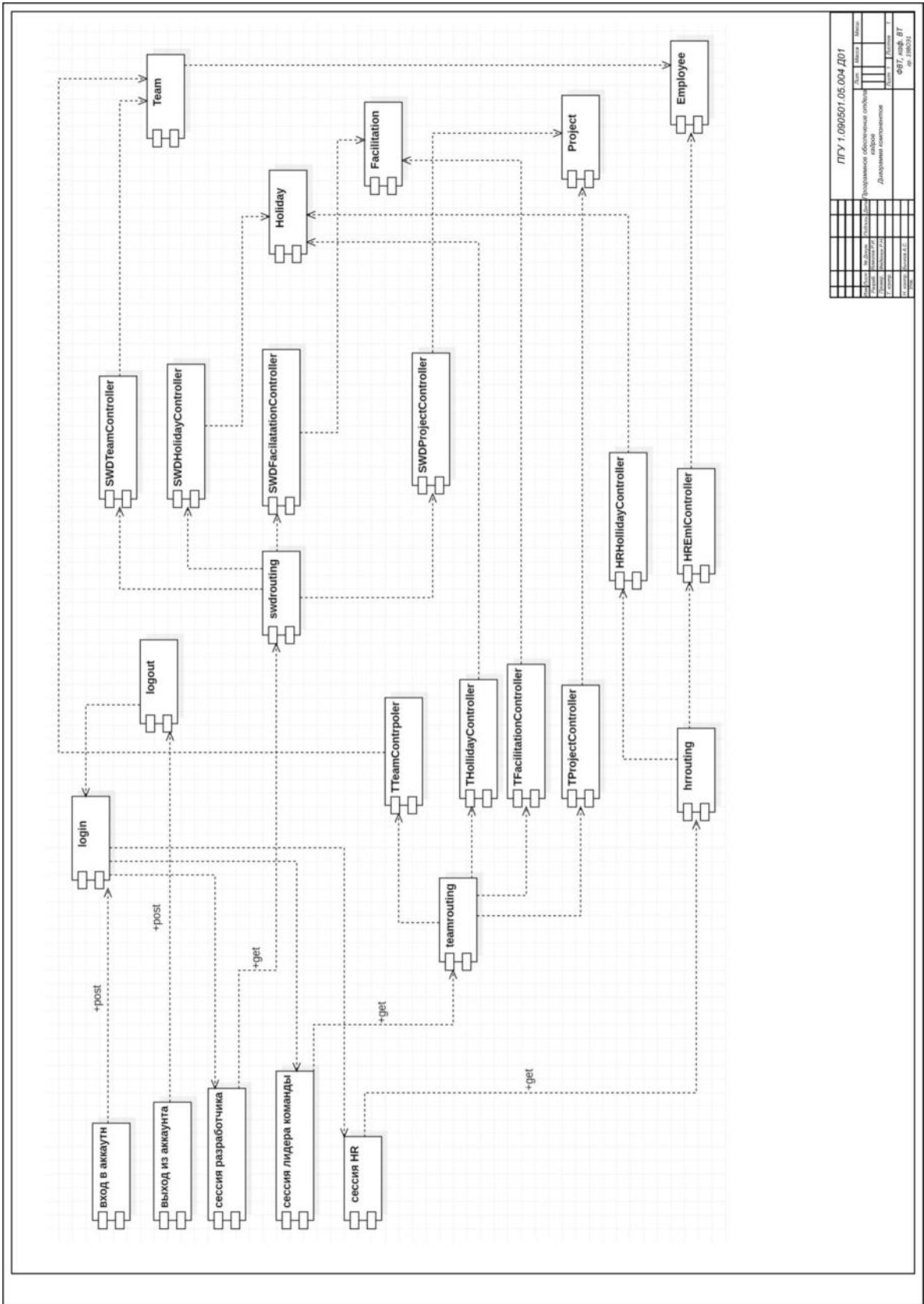
Приложение А.2 – Диаграмма развертывания



Приложение А.3 – Логическая модель данных

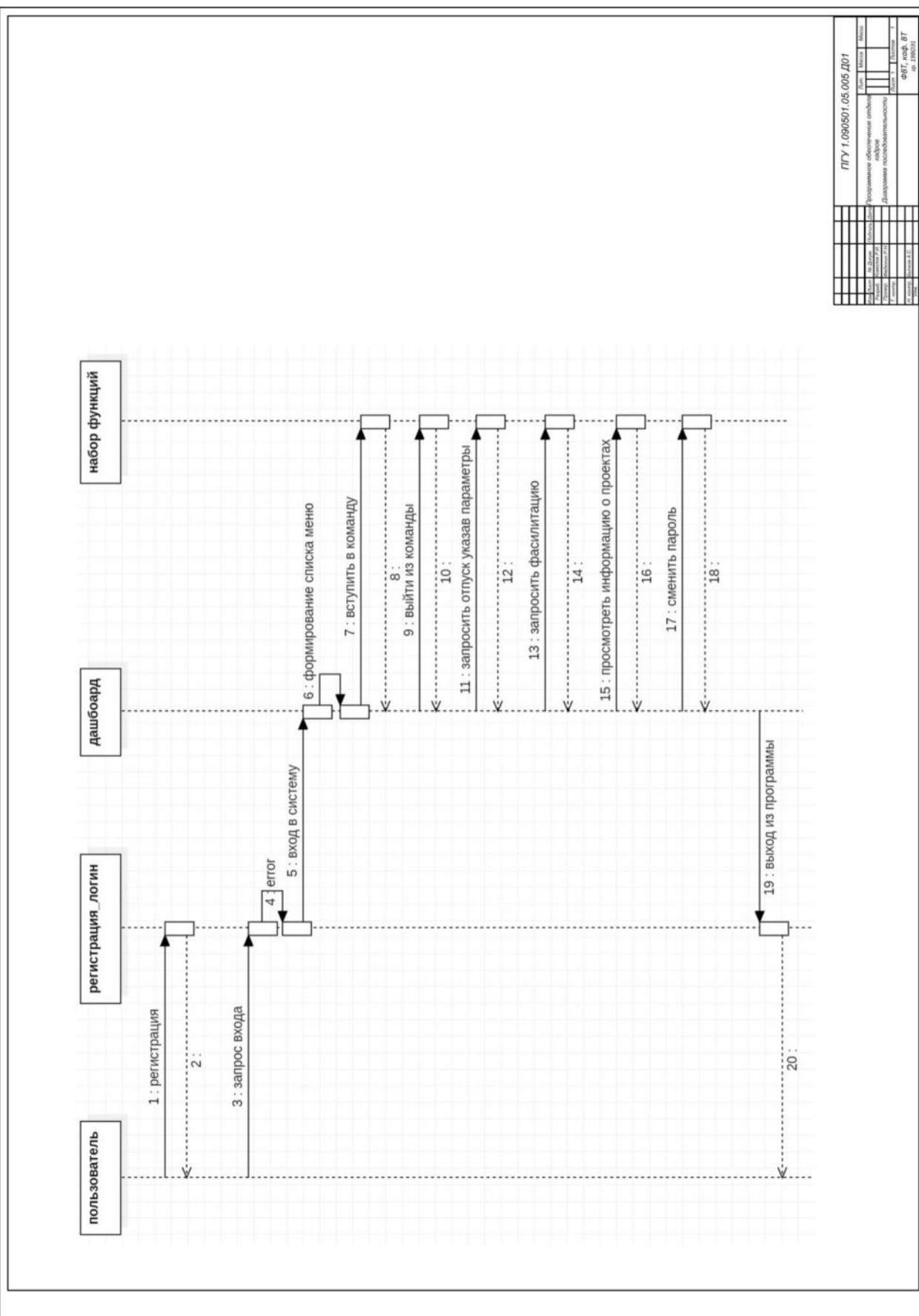


Приложение А.4 – Диаграмма компонентов

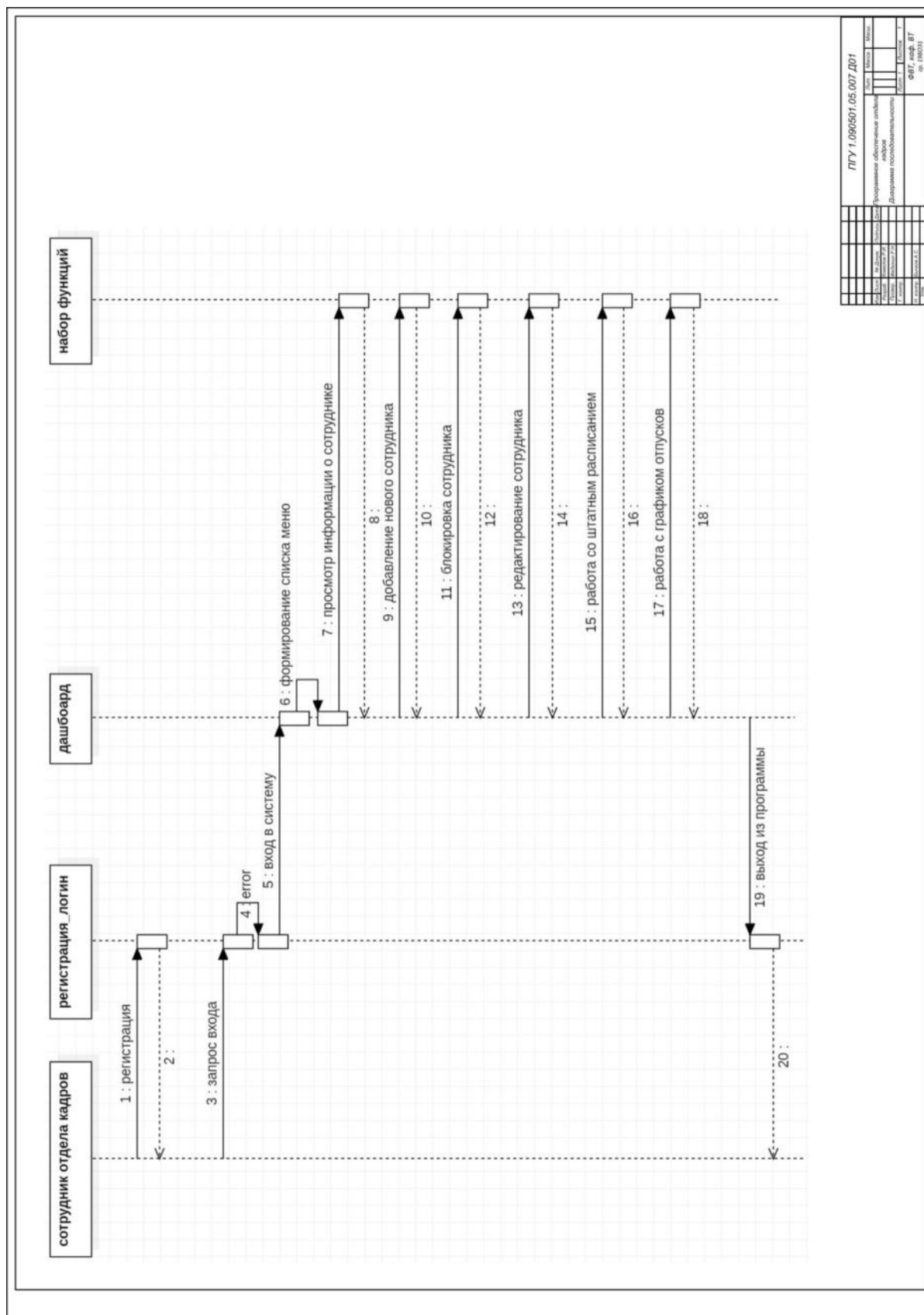


ПУ 1.090501.05.004.001	Имя	Место
Программное обеспечение системы	Имя	Место
Диаграмма компонентов	Имя	Место
ФЭТ, табл. БТ	Имя	Место
ИД: 1090501	Имя	Место

Приложение А.5 – Диаграмма последовательности (разработчик)

[illegible]

Приложение А.6 – Диаграмма последовательности (лидер команды)



кадров)



Приложение Б

(обязательное)

Программное обеспечение отдела кадров

Листинг программы

					ПГУ 1.090501.05.001 ПЗ	Лист
						81
Изм.	Лист	№ докум.	Подпись	Дата		

Модуль index.php

```
<?php
    $result = "";
    if(isset($_GET['msg']))
    {
        $result=$_GET['msg'];
    }
?>

<!DOCTYPE HTML>
<html>
<head>
<title>Авторизация</title>
<meta name="viewport" content="width=device-width, initial-scale=1">
<meta http-equiv="Content-Type" content="text/html; charset=utf-8"
/>
<meta name="keywords" content=""
/>
<script type="application/x-javascript"> addEventListener("load",
function() { setTimeout(hideURLbar, 0); }, false); function
hideURLbar(){ window.scrollTo(0,1); } </script>
<!-- Bootstrap Core CSS -->
<link href="css/bootstrap.min.css" rel='stylesheet' type='text/css'
/>
<!-- Custom CSS -->
<link href="css/style.css" rel='stylesheet' type='text/css' />
<link rel="stylesheet" href="css/morris.css" type="text/css"/>
<!-- Graph CSS -->
<link href="css/font-awesome.css" rel="stylesheet">
<link rel="stylesheet" href="css/jquery-ui.css">
<!-- jQuery -->
<script src="js/jquery-2.1.4.min.js"></script>
<!-- //jQuery -->
```

```

<link
href='//fonts.googleapis.com/css?family=Roboto:700,500,300,100italic
,100,400' rel='stylesheet' type='text/css'/>
<link href='//fonts.googleapis.com/css?family=Montserrat:400,700'
rel='stylesheet' type='text/css'>
<!-- lined-icons -->
<link rel="stylesheet" href="css/icon-font.min.css" type='text/css'
/>
<!-- //lined-icons -->
<style>
    html{
        min-height: calc(100%);
        width:calc(100%);
    }
    body, .main-wthree{
        width:calc(100%);
        min-height: 100vh;
    }
    .main-wthree{
        padding-bottom:2em;
        display:flex;
        flex-direction: column;
        align-items:center;
        justify-content:center;
    }
    .footer{
        width:100%;
        position:fixed;
        bottom:0;
        left:0
    }
    .sin-w3-agile{
        padding:0;
    }
    .login{

```

```

        background-color: #010101;
        background-image: linear-gradient(160deg, #010101 0%, #4e6865
100%);
    }
    .login-w3 {
        width: 100%;
        float: unset;
        text-align: center;
    }
    .main-wthree input[type="submit"]:hover {
        background: #3e5250;
    }
</style>
</head>
<body>
    <div class="main-wthree">
        <div class="container">
            <h1 class="text-center text-white">Система управления
человеческими ресурсами</h1>
            <div class="sin-w3-agile">
                <h2>Авторизация</h2>
                <form action="controller/login.php" method="post">
                    <div class="email">
                        <span class="email">Почта:</span>
                        <input type="Email" name="name" class="name"
placeholder="Введите адрес электронной почты">
                        <div class="clearfix"></div>
                    </div>
                    <div class="password-agileits">
                        <span class="username">Пароль: <i class="fa
fa-eye-slash" aria-hidden="false" style="padding-left: 20px;"
onclick="passwordeyes(this);"></i></span>
                        <input type="password" name="password"
id="Psw" class="password" placeholder="Введите пароль">
                        <div class="clearfix"></div>

```

```

        </div>
        <h4 style="color: #F1C40F;"><?php echo $result;
?></h4>

        <div class="login-w3">
            <input type="submit" name="submit"
class="login" value="Войти">
        </div>
        <div class="clearfix"></div>
        <h5 class="text-center"><a href="./user"
class="text-white" >Войти как сотрудник</a></h5>
        <div class="clearfix"></div>
    </form>

        <!-- <div class="back">
            <a href="index.php">Back to home</a>
        </div> -->
        <div class="footer">
            <p>Система управления человеческими
ресурсами. Все права защищены &copy; <?= date("Y") ?> </p>
        </div>

    </div>
</div>

</div>

</div>
<script>
function passwordeyes(_this) {
    var x = document.getElementById("Psw").type;
    if(x=="password"){
        document.getElementById("Psw").type="text";
        _this.setAttribute('class', "fa fa-eye")
    }else{
        document.getElementById("Psw").type="password";
        _this.setAttribute('class', "fa fa-eye-slash")
    }
}
</script>

```

```
</body>
```

```
</html>
```

```
<?php
```

```
?>
```

Модуль dashboard.php

```
<?php include('header.php'); ?>
```

```
<?php
```

```
include_once('controller/connect.php');
```

```
$dbs = new database();
```

```
$db=$dbs->connection();
```

```
$TotalEmp =mysqli_query($db,"select count(EmployeeId) as emp from  
employee where RoleId !='1' ");
```

```
$TotalEmploId = mysqli_fetch_assoc($TotalEmp);
```

```
$spandingleave = mysqli_query($db,"select count(LeaveStatus) as  
pleave from leavedetails where LeaveStatus='Pending'");
```

```
$tpandingleave = mysqli_fetch_assoc($spandingleave);
```

```
?>
```

```
<ol class="breadcrumb" style="margin: 10px 0px ! important;">
```

```
    <li class="breadcrumb-item" title="Home"><a  
href="index.php">Главная</a></li>
```

					ПГУ 1.090501.05.001 ПЗ	Лист
						86
Изм.	Лист	№ докум.	Подпись	Дата		

```

</ol>

<!--four-grids here-->

    <div class="row" style="">

        <div class="col-lg-6 col-md-6 col-sm-12">

            <div class="panel panel-default bg-
light dash-summary">

                <div class="panel-body">

                    <a href="employeeview.php">

                        <div class="icon">

                            <i class="glyphicon
glyphicon-user" aria-hidden="true"></i>

                        </div>

                        <div class="four-text">

                            <h3>Сотрудники</h3>

                            <h4><?php
echo(isset($TotalEmploId['emp']))?$TotalEmploId['emp']:"";?></h4>

                        </div>

                    </a>

                </div>

            </div>

        </div>

    <div class="col-lg-6 col-md-6 col-sm-12">

        <div class="panel panel-default bg-
light dash-summary">

```

```

<div class="panel-body">

    <a href="leaverequest.php">

        <div class="icon">

            <i class="glyphicon
glyphicon-list-alt" aria-hidden="true"></i>

        </div>

        <div class="four-text">

<h3>Посещаемость</h3>

                                <h4><?php
echo(isset($stpandingleave['pleave']))?$stpandingleave['pleave']:"";?>
</h4>

                                </div>

                                </a>

                                </div>

                                </div>

                                </div>

                                <!-- <div class="col-md-3 four-grid">

                                    <div class="four-w3ls">

                                        <div class="icon">

                                            <i class="glyphicon
glyphicon-folder-open" aria-hidden="true"></i>

                                        </div>

                                        <div class="four-text">

```

<h3>Projects</h3>

<h4>12,430</h4>

</div>

</div>

</div> -->

<!-- <div class="col-md-3 four-grid">

<div class="four-wthree">

<div class="icon">

<i class="glyphicon
glyphicon-briefcase" aria-hidden="true"></i>

</div>

<div class="four-text">

<h3>Old Projects</h3>

<h4>14,430</h4>

</div>

</div>

</div> -->

<div class="clearfix"></div>

</div>

<!--//four-grids here-->

<?php include('footer.php'); ?>

Модуль EmplController.php

<?php

include_once('connect.php');

\$dbs = new database();

\$db=\$dbs->connection();

session_start();

if(isset(\$_POST['submit']))

{

 \$data=\$_POST;

 \$editid = 0;

 if(isset(\$_GET['empedit']) && \$_GET['empedit'] > 0){

 \$editid = \$_GET['empedit'];

 };

 \$empid=\$data['empid'];

 \$img=\$_FILES['pfimg']['name'];

 \$gender=\$data['gender'];

 \$fname=\$data['fname'];

 \$mname=\$data['mname'];

 \$lname=\$data['lname'];

 \$bdate=\$data['bdate'];

| | | | | | | |
|------|------|----------|---------|------|------------------------|------|
| | | | | | ПГУ 1.090501.05.001 ПЗ | Лист |
| | | | | | | 90 |
| Изм. | Лист | № докум. | Подпись | Дата | | |

```

$mnumber=$data['mnumber'];

$email=$data['email'];

$address1=$data['address1'];

$address2=$data['address2'];

$address3=$data['address3'];

$city=$data['city'];

$joindate=$data['joindate'];

$leavedate=$data['leavedate'];

$status=$data['status'];

$role=$data['role'];

$password=$data['password'];

$marital=$data['marital'];

$position=$data['position'];

$imagefilename = $data['imagefilename'];

$imageComplete=false;

if($editid==0){

    $sql = mysqli_query($db,"select * from employee where
Email='$email'");

}

else{

    $sql = mysqli_query($db,"select * from employee where
Email='$email' and empid!=$editid");

```

```

    }

    if(mysqli_num_rows($sql) > 0)

    {

        header("location:../employeeadd.php?msg=Email address
all ready existed!");exit;

    }

    else

    {

        if(!empty($_FILES['pfimg']['name']))

        {

            $name=$_FILES['pfimg']['name'];

            $temp=$_FILES['pfimg']['tmp_name'];

            $size=$_FILES['pfimg']['size'];

            $type=$_FILES['pfimg']['type'];

            if($type != "image/jpg" && $type != "image/png"
&& $type != "image/jpeg" && $type != "image/gif")

            {

                header("location:../employeeadd.php?msg=Invalid image !");exit;

            }

            else

```

```

        {

            if($size > 1000000)

                {

                    header("location:../employeeadd.php?msg=File size upto 1MB
required ! ");exit;

                }

            else

                {

                    $ImageComplete=true;

                }

        }

    }

else

    {

        $in = $_POST["imagefilename"];

        if(file_exists("../image/".$in))

            {

                $ImageComplete=true;

            }

        else

            {

```

```

        header("location:../employeeadd.php?msg=Pleaes
Select Profile Image! ");exit;

    }

}

}

if($ImageComplete)

{

    $roleid = $_SESSION['User']['RoleId'];

    date_default_timezone_set("Asia/Kolkata");

    $datetime = date("Y-m-d h:i:s");

    if($editid==0)

    {

        if(!empty($_FILES['pfimg']['name']))

        {

            $name = rand(222,333333).$name;

            move_uploaded_file($temp,"../image/".$name);

        }

        else

        {

            $name = $_POST["imagefilename"];

        }

    }

    mysqli_query($db,"insert into employee
values(null,'$empid','$fname','$mname','$lname','$bdate','$gender','

```

| | | | | | | |
|------|------|----------|---------|------|------------------------|------|
| | | | | | ПГУ 1.090501.05.001 ПЗ | Лист |
| | | | | | | 94 |
| Изм. | Лист | № докум. | Подпись | Дата | | |

```
$address1','$address2','$address3','$city','$mnumber','$email','$password','$aadharcard','$marital','$position','$roleid','$datetime',null,null,'$joindate','$leavedate',null,null,'$status','$role','$name','$macaddress')");
```

```
header("location:../detailview.php?id=Successfull... ");exit;
```

```
}
```

```
else
```

```
{
```

```
if(!empty($_FILES['pfimg']['name']))
```

```
{
```

```
$name = rand(222,333333).$name;
```

```
move_uploaded_file($temp,"../image/".$name);
```

```
}
```

```
else
```

```
{
```

```
$name = $_POST["imagefilename"];
```

```
}
```

```
mysqli_query($db,"update employee set
```

```
EmployeeId='$empid',FirstName='$fname',MiddleName='$mname',LastName='$lname',Birthdate='$bdate',Gender='$gender',Address1='$address1',Address2='$address2',Address3='$address3',CityId='$city',Mobile='$mnumber',Email='$email',Password='$password',AadharNumber='$aadharcard',MaritalStatus='$marital',PositionId='$position',ModifiedBy='$roleid',ModifiedDate='$datetime',JoinDate='$joindate',LeaveDate='$leavedate',StatusId='$status',RoleId='$role',ImageName='$name',MacAddress='$macaddress' where EmpId='$editid' ");
```

| | | | | | | |
|------|------|----------|---------|------|------------------------|------|
| | | | | | ПГУ 1.090501.05.001 ПЗ | Лист |
| | | | | | | 95 |
| Изм. | Лист | № докум. | Подпись | Дата | | |

```

        header("location:../detailview.php?employeeid=$editid");exit;

    }

}

}

?>

```

Модуль TeamController.php

```

<?php
    include_once('connect.php');
    session_start();

    $dbs = new database();
    $db=$dbs->connection();
    $positionedit="";
    if(isset($_POST['cityy']))
    {
        $cityid = $_GET['cityedit'];
        $state = $_POST['state'];
        $city = $_POST['city'];
        if(is_numeric($cityid) && $cityid > 0){
            $sql = mysqli_query($db,"select * from city where
CityId='$cityid'");
            if(mysqli_num_rows($sql))
            {
                mysqli_query($db,"update city set
StateId='$state',Name='$city' where CityId='$cityid'");
                echo
"<script>window.location='../city.php';</script>";
            }
        }
    }
}

```

| | | | | | | |
|------|------|----------|---------|------|------------------------|------|
| | | | | | ПГУ 1.090501.05.001 ПЗ | Лист |
| | | | | | | 96 |
| Изм. | Лист | № докум. | Подпись | Дата | | |

```

        }else
        {
            mysqli_query($db,"insert into city
values(null,'$state','$city')");
            echo
"<script>window.location='../city.php';</script>";
        }
    }
    else if(isset($_POST['addstates']))
    {
        $stateid = $_GET['stateedit'];
        $country = $_POST['country'];
        $state = $_POST['state'];
        if(is_numeric($stateid) && $stateid > 0){
            $sql = mysqli_query($db,"select * from state
where StateId='$stateid'");
            if(mysqli_num_rows($sql))
            {
                mysqli_query($db,"update state set
CountryId='$country',Name='$state' where StateId='$stateid'");
                echo
"<script>window.location='../state.php';</script>";
            }
        }else
        {
            mysqli_query($db,"insert into state
values(null,'$country','$state')");
            echo
"<script>window.location='../state.php';</script>";
        }
    }

    else if(isset($_POST['countryy']))
    {
        $countryid = $_GET['countryedit'];

```

```

        $countryn = $_POST['country'];
        if(is_numeric($countryid) && $countryid > 0){
            $sql = mysqli_query($db,"select * from country
where CountryId='$countryid'");
            if(mysqli_num_rows($sql))
            {
                mysqli_query($db,"update country set
Name='$countryn' where CountryId='$countryid'");
                echo
"<script>window.location='../country.php';</script>";
            }
        }else
        {
            mysqli_query($db,"insert into country
values(null,'$countryn')");
            echo
"<script>window.location='../country.php';</script>";
        }
    }
    else if(isset($_POST['positionl']))
    {
        $positionid = $_GET['positionedit'];
        $positionn = $_POST['position'];
        if(is_numeric($_GET['positionedit']) &&
$_GET['positionedit'] > 0){
            $sql = mysqli_query($db,"select * from position
where PositinId='$positionid'");
            if(mysqli_num_rows($sql))
            {
                mysqli_query($db,"update position set
Name='$positionn' where PositinId='$positionid' ");
                echo
"<script>window.location='../position.php';</script>";
            }
        }
    }
}

```

```

        else
        {
            mysqli_query($db,"insert into position
values(null,'$positionn')");
            echo
"<script>window.location='../position.php';</script>";
        }
    }
    else if(isset($_GET['statedelete']))
    {
        $delet = $_GET['statedelete'];

        mysqli_query($db,"delete from state where
StateId='$delet'");
        echo
"<script>window.location='../state.php';</script>";
    }
    else if(isset($_GET['citydelete']))
    {
        $delet = $_GET['citydelete'];

        mysqli_query($db,"delete from city where
CityId='$delet'");
        echo
"<script>window.location='../city.php';</script>";
    }
    else if(isset($_GET['countrydelete']))
    {
        $delet = $_GET['countrydelete'];

        mysqli_query($db,"delete from country where
CountryId='$delet'");
        echo
"<script>window.location='../country.php';</script>";
    }

```

| | | | | |
|------|------|----------|---------|------|
| | | | | |
| Изм. | Лист | № докум. | Подпись | Дата |

ПГУ 1.090501.05.001 ПЗ

Лист

99

```

else if(isset($_GET['positiondelete']))
{
    $delet = $_GET['positiondelete'];

    mysqli_query($db,"delete from position where
PositinId='$delet'");
    echo
"<script>window.location='../position.php';</script>";
}
else
{
}
?>

```

| | | | | | | |
|------|------|----------|---------|------|------------------------|------|
| | | | | | ПГУ 1.090501.05.001 ПЗ | Лист |
| | | | | | | 100 |
| Изм. | Лист | № докум. | Подпись | Дата | | |

Приложение В

(обязательное)

Программное обеспечение отдела кадров

Презентация

| | | | | | | |
|------|------|----------|---------|------|-------------------------------|------|
| | | | | | <i>ПГУ 1.090501.05.001 ПЗ</i> | Лист |
| | | | | | | 101 |
| Изм. | Лист | № докум. | Подпись | Дата | | |

ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ ОТДЕЛА КАДРОВ

Автор работы:

студент гр. 19ВОЭ1 Комолов Роман Игоревич

Научный руководитель:

к.т.н., доцент Федюнин Роман Николаевич

Пенза, 2024

| | | | | |
|------|------|----------|---------|------|
| | | | | |
| Изм. | Лист | № докум. | Подпись | Дата |

ПГУ 1.090501.05.001 ПЗ

Цель работы и объекты

Цель работы – реализовать программное приложение предназначенное для организации команд разработчиков, с поддержкой данной работы отделом кадров.

Приложение необходимо для трех категорий пользователей:

- Разработчики – пользователи с правом работы в проектных командах и правом голоса;
- Лидеры команд – пользователи с правом формирования и управления командами.
- Сотрудники отдела кадров – пользователи с правом работы со штатным расписанием организации.

Проект является клиент-серверным приложением, выполняющий функции:

- Организация команд.
- Организация времени работы и отдых сотрудников.
- Организация совещаний.
- Организация штатного состава команд.

| | | | | |
|------|------|----------|---------|------|
| | | | | |
| Изм. | Лист | № докум. | Подпись | Дата |

ПГУ 1.090501.05.001 ПЗ

Задачи приложения

предоставить разработчику возможность отслеживать его траекторию профессионального развития.

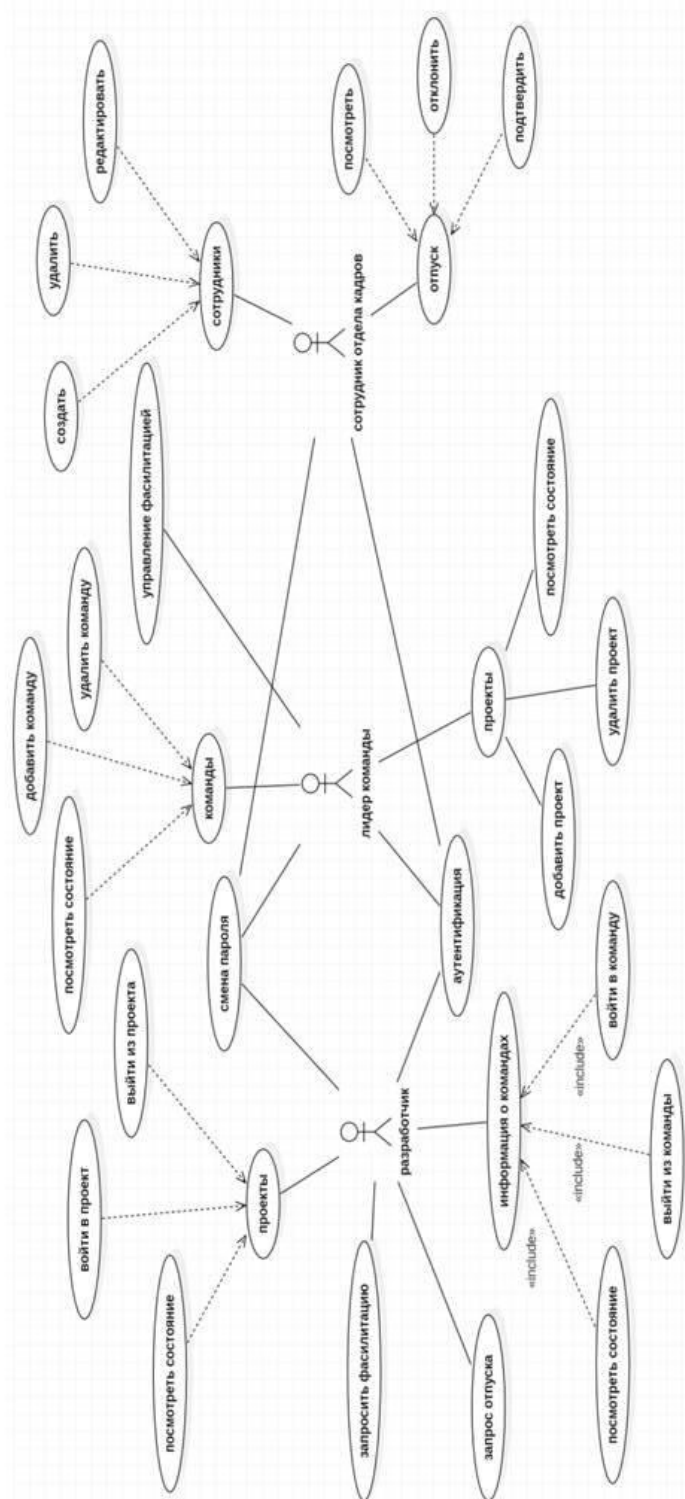
предоставить лидеру команд возможность работы с командами, составом команд и с моральным духом коллектива, посредством фасилитации.

предоставить сотруднику отдела кадров работать с сотрудниками и штатным расписанием.

| | | | | |
|------|------|----------|---------|------|
| | | | | |
| Изм. | Лист | № докум. | Подпись | Дата |

ПГУ 1.090501.05.001 ПЗ

Диаграмма вариантов использования



| | | | | |
|------|------|----------|---------|------|
| | | | | |
| | | | | |
| Изм. | Лист | № докум. | Подпись | Дата |

Сравнение аналогов

1. BambooHR: облачный сервис
2. Comindware: предлагает систему управления HR-процессами
3. SAP Staff Central: комплексная HRIS
4. 1С Кадры предлагает широкий спектр функций для управления HR-процессами

При этом фактически все решения лишены функций работы с психологическим фактором работы в команде и фасилитацией команд.

| | | | | |
|------|------|----------|---------|------|
| | | | | |
| Изм. | Лист | № докум. | Подпись | Дата |

ПГУ 1.090501.05.001 ПЗ

Стек технологий

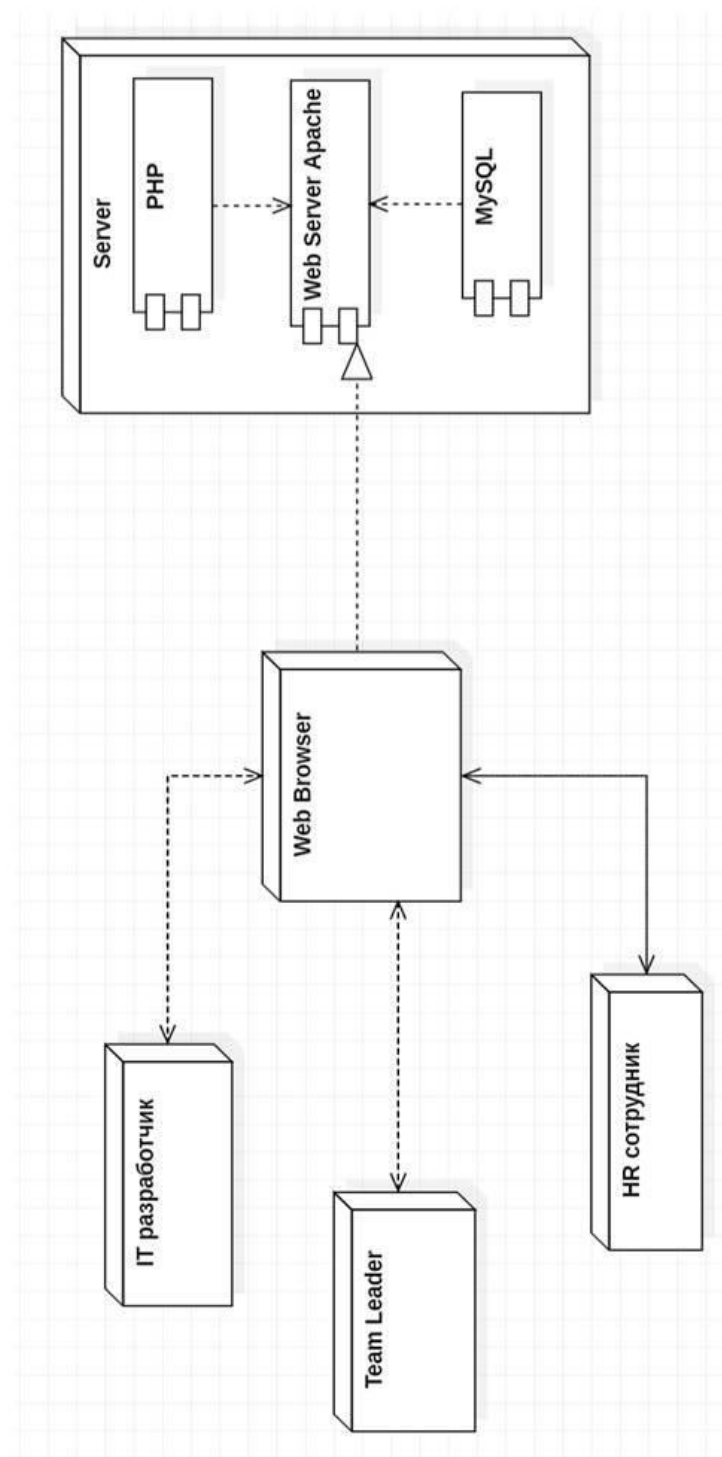


| | | | | |
|------|------|----------|---------|------|
| | | | | |
| Изм. | Лист | № докум. | Подпись | Дата |

ПГУ 1.090501.05.001 ПЗ

Лист
107

Архитектура приложения



Структура базы данных

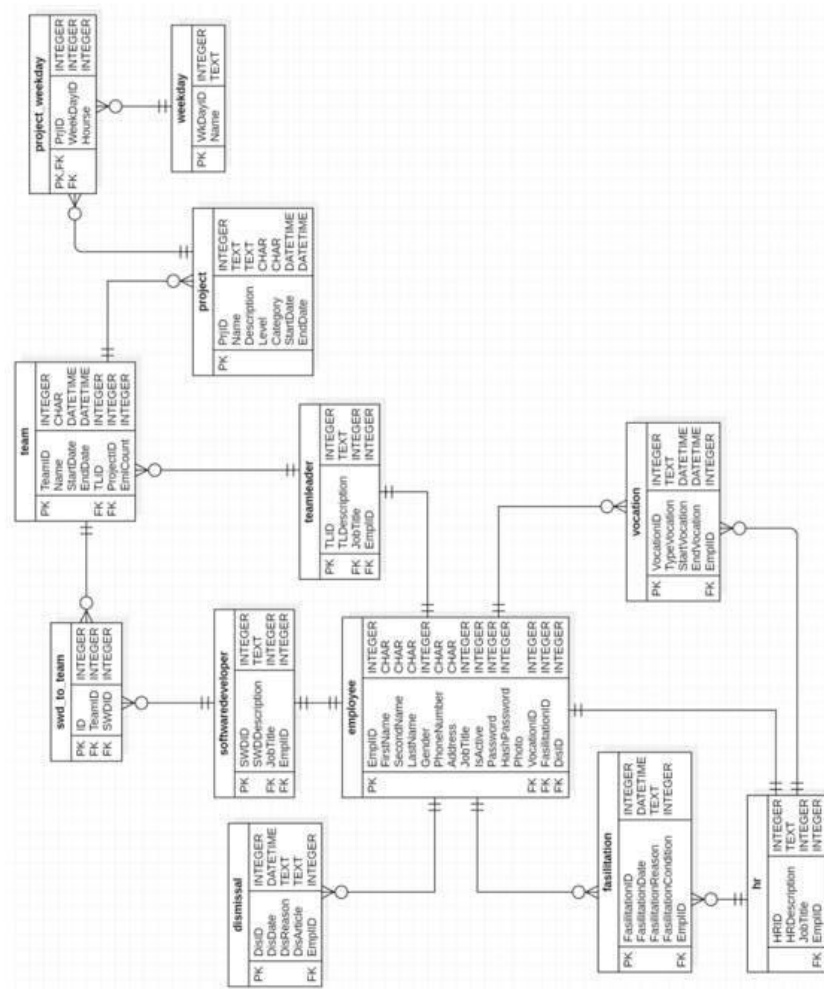


Диаграмма компонентов

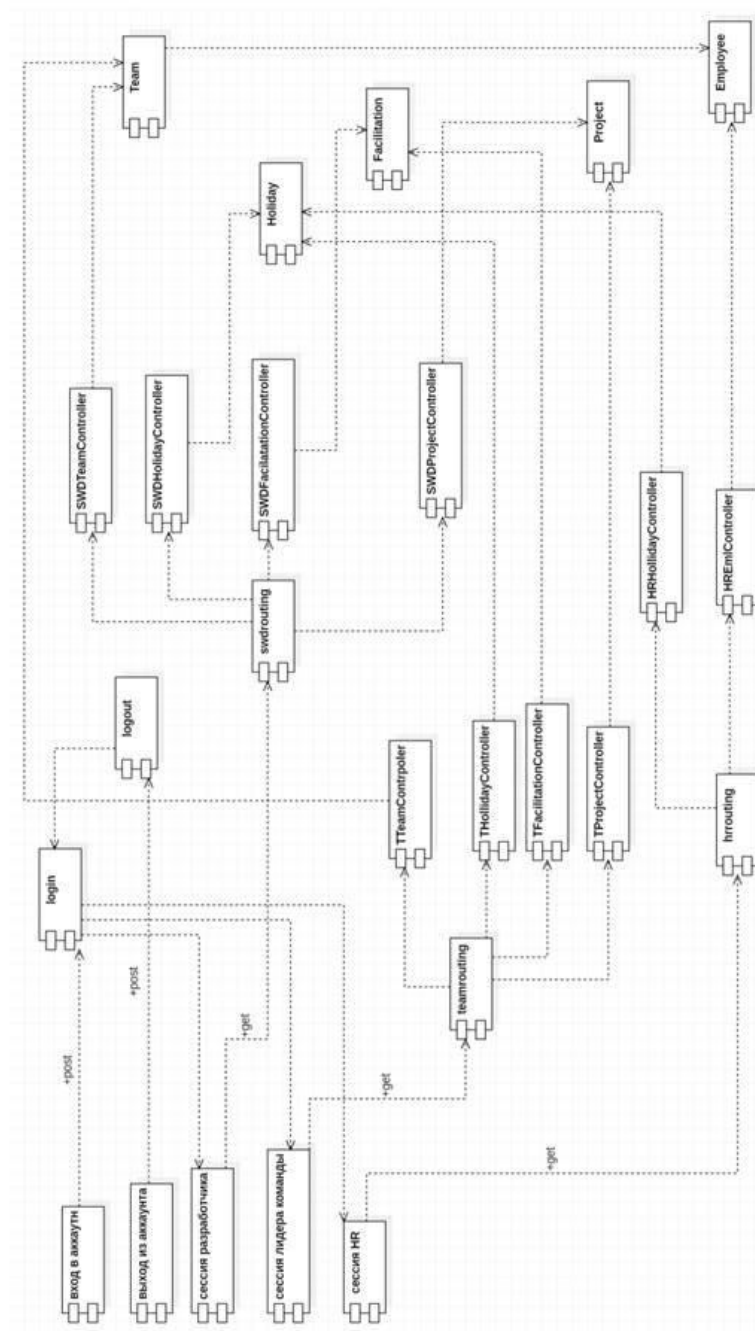


Диаграмма последовательности (разработчик)

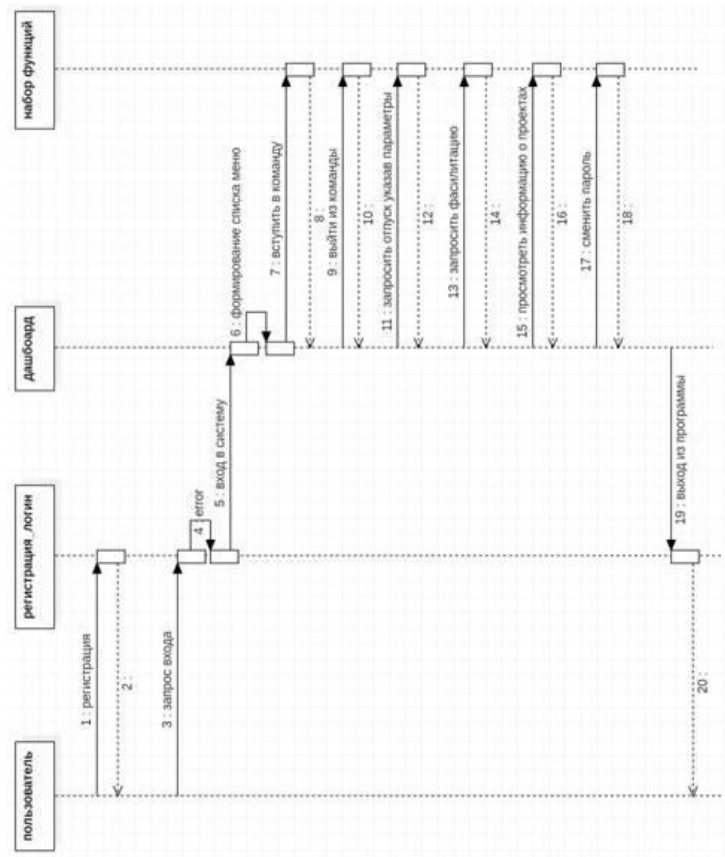


Диаграмма последовательности (лидер команды)

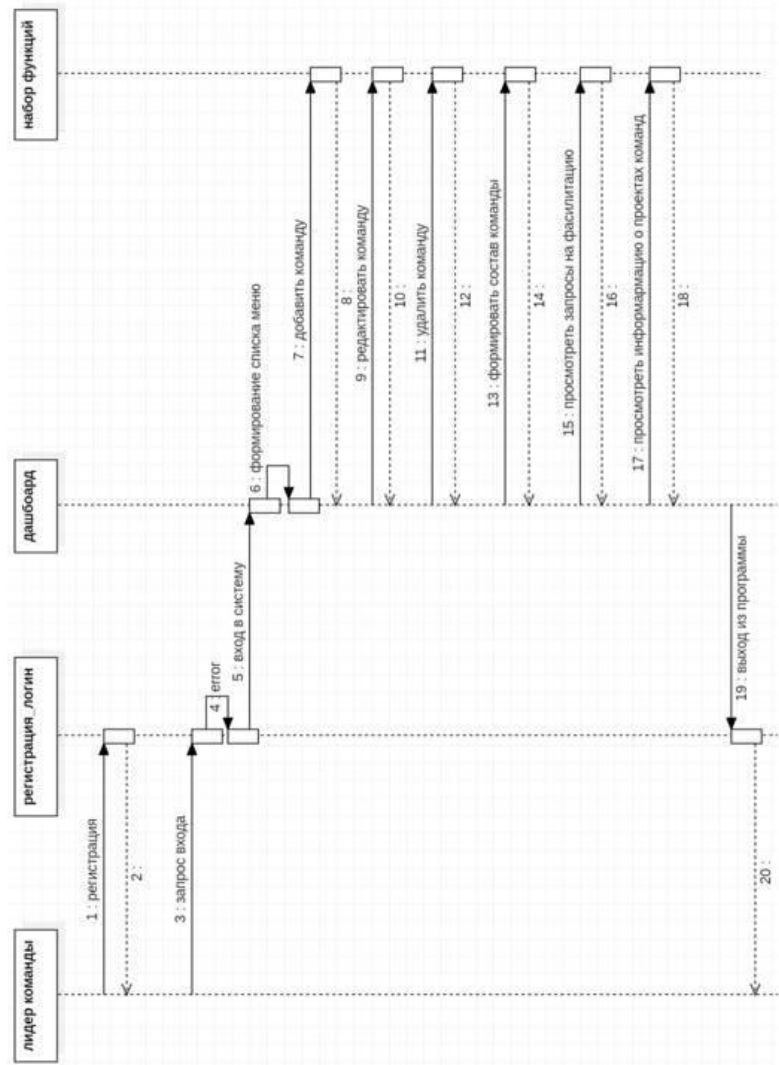
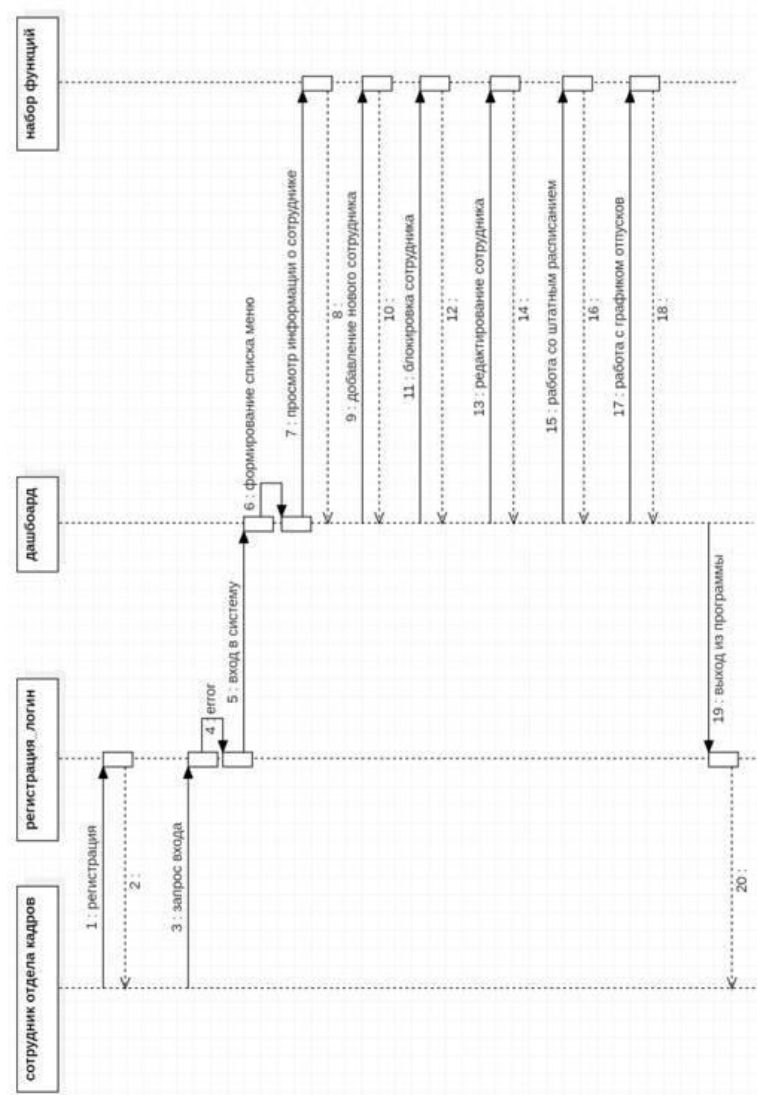
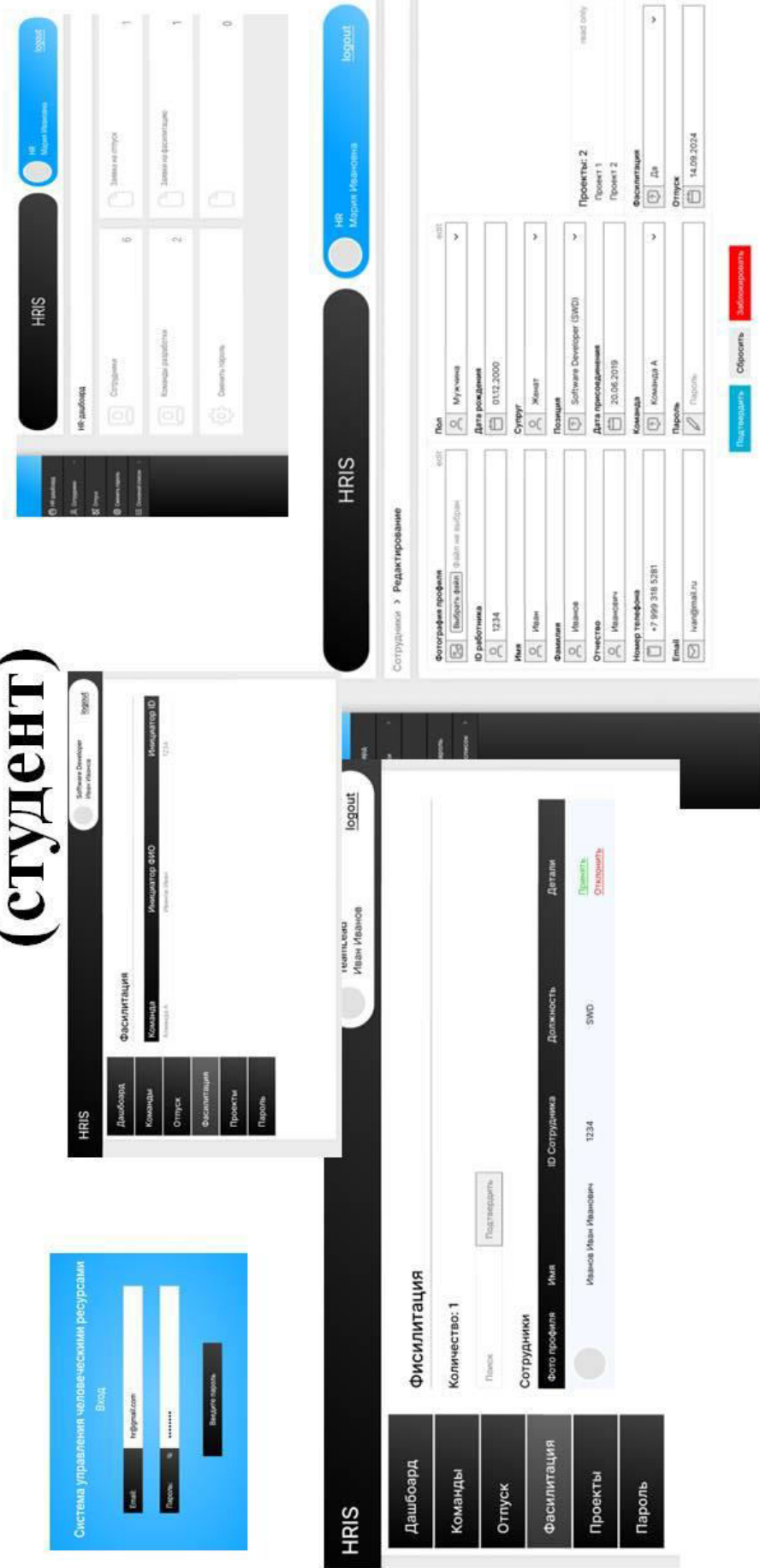


Диаграмма последовательности (сотрудник отдела кадров)



Скриншоты запущенного приложения (студент)



Результаты работы

- Разработано приложение полностью соответствующее техническому заданию;
- Локально настроена техническая среда для исполнения приложения;
- Проведено функциональное тестирование (успешно);
- Намечены задачи по развитию функциональных возможностей программы.

| | | | | |
|------|------|----------|---------|------|
| | | | | |
| Изм. | Лист | № докум. | Подпись | Дата |

ПГУ 1.090501.05.001 ПЗ

Спасибо за внимание!

| | | | | | | |
|------|------|----------|---------|------|------------------------|------|
| | | | | | ПГУ 1.090501.05.001 ПЗ | Лист |
| | | | | | | 116 |
| Изм. | Лист | № докум. | Подпись | Дата | | |