

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РЕСПУБЛИКИ КАЗАХСТАН
Некоммерческое акционерное общество
«АЛМАТИНСКИЙ УНИВЕРСИТЕТ ЭНЕРГЕТИКИ И СВЯЗИ
ИМЕНИ ГУМАРБЕКА ДАУКЕЕВА»
Кафедра «IT инжиниринг»

ДОПУЩЕН К ЗАЩИТЕ

Зав.кафедрой PhD, доцент Досжанова А.А.

_____ « ____ » _____ 2020 г.
(подпись)

ДИПЛОМНЫЙ ПРОЕКТ

На тему: Разработка информационной системы проектно-строительной организации

Специальность 5В070300 – «Информационные системы»

Выполнила Черткова Н.О. Группа ИС-16-2

Научный руководитель д.т.н., профессор Ахметов Б.С.

Консультанты:

по экономической части: к.э.н., асоц. профессор Габелашвили К.Р.

_____ « ____ » _____ 2020 г.
(подпись)

по безопасности жизнедеятельности: к.т.н. доцент Приходько Н.Г.

_____ « ____ » _____ 2020
г.
(подпись)

по применению вычислительной техники: магистр ст. преп. Майкотов М.Н.

_____ « ____ » _____ 2020
г.
(подпись)

Нормоконтролер: магистр ст. преп. Абсатарова Б.Р.

_____ « ____ » _____ 2020
г.
(подпись)

Рецензент: _____
(ученая степень, звание, Ф.И.О.)

Г. _____ « ____ » _____ 2020
(подпись)

Алматы 2020
МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РЕСПУБЛИКИ КАЗАХСТАН
Некоммерческое акционерное общество
«АЛМАТИНСКИЙ УНИВЕРСИТЕТ ЭНЕРГЕТИКИ И СВЯЗИ
ИМЕНИ ГУМАРБЕКА ДАУКЕЕВА»

Институт систем управления и информационных технологий

Кафедра «IT инжиниринг»

Специальность 5В070300 – «Информационные системы»

ЗАДАНИЕ
на выполнение дипломного проекта

Студенту Черткова Наталья Олеговна

Тема работы: Разработка информационной системы проектно-строительной организации

Утверждена приказом по университету № _____ от «__» _____ 2020 г.

Срок сдачи законченного проекта «____» _____ 2020 г.

Исходные данные к проекту (требуемые параметры результатов исследования (проектирования) и исходные данные объекта): веб-приложения и системы по управлению проектами, документации языков программирования.

Перечень вопросов, подлежащих разработке в дипломном проекте, или краткое содержание дипломного проекта:

а) анализ современных систем управления проектами и методов управления;

б) программная реализация веб-приложения;

в) вопросы безопасности жизнедеятельности и охраны труда;

г) экономическая эффективность.

Перечень графического материала (с точным указанием обязательных чертежей): 16 таблиц, 70 иллюстраций.

Основная рекомендуемая литература:

1 ДеМарко Т., Листер Т. Человеческий фактор. Успешные проекты и команды СПб: Символ-Плюс, 2011. – 256с.

2 Филипова О. Изучаем Vue.js Бирмингем-Мумбаи: Packt Publishing, 2016. – 323с.

3 Скляр Д. Изучаем PHP 7: руководство по созданию интерактивных веб-сайтов. : Пер. с англ. — СПб.: ООО “Альфа-книга”, 2017. — 464 с.

4 Крокфорд Д. JavaScript: Сильные стороны – СПб.: Питер, 2012. – 176с.

Консультации по проекту с указанием относящихся к ним разделов проекта

Раздел	Консультант	Сроки	Подпись
Экономическая часть	Габелашвили К.Р	5.04.2020 – 23.04.2020	
Безопасность жизнедеятельности	Приходько Н.Г.	3.04.2020 – 22.04.2020	
Программная часть	Майкотов М.Н.	4.05.2020 – 14.05.2020	
Нормоконтролер	Абсатарова Б.Р.	4.05.2020 – 13.05.2020	

График
подготовки дипломного проекта

Наименование разделов, перечень разрабатываемых вопросов	Сроки представления научному руководителю	Примечание
Аналитическая часть	13.01.2020 – 17.02.2020	
Разработка информационного обеспечения системы	19.02.2020 – 15.03.2020	
Разработка программного обеспечения системы	20.03.2020 – 3.05.2020	

Дата выдачи задания « ___ » _____ 2020 г.

Заведующий кафедрой _____ Досжанова А.А.
(подпись)

Научный руководитель проекта _____ Ахметов Б.С.
(подпись)

Задание принял к исполнению студент _____ Черткова Н.О.
(подпись)

Аңдатпа

«Жобалау мен құрылысты ұйымдастырудың ақпараттық жүйесі» жобалау мен ұйымның жобалық тобының жұмысын басқару процесін автоматтандыру үшін жасалды.

Бұл жүйе бір парақты кросс-платформасы және бейімделген қосымшасы негізінде жүзеге асырылады. Жобаларды басқарудың заманауи әдістерін талдау барысында ұйымдарда жобаларды басқарудың ең қолайлы әдістері Kanban and Scrum екендігі белгілі болды. Осы басқару әдістерінің негізінде ұйымның жеке талаптарына сәйкес келетін өзіндік әдістеме құрылды.

Канбан әдісі бойынша ұқсас жүйелерді талдау қазіргі заманғы жобаларды басқару жүйелері өте жүктелген және интуитивті емес интерфейске ие, ұйымның жеке талаптарына сәйкес келмейді, шамадан тыс функционалдылыққа ие және кейбір веб-қызметтерді зерттеуге уақыт қажет екенін көрсетті жоба менеджменті. Сондықтан ұйым үшін ең оңтайлы шешім жеке веб-қосымшаны құру болды.

Жүйеге қойылатын басты талап әр түрлі жастағы және компьютерде жұмыс істейтін әр түрлі қызметкерлер үшін ыңғайлы қарапайым және интуитивті интерфейс болды.

Дипломдық жобаны жазудың нәтижесі – бұл жобалық топтың жұмысын басқаруға мүмкіндік беретін кросс-платформалық және бейімделетін бір парақты веб-қосымшасы, басымдылықты міндеттерді көзбен көру және уақытты дұрыс басқару арқылы қызметкерлердің сапасы мен жылдамдығын жақсарту.

Әзірленген қосымша ұйымның тиімділігін арттыруға, сонымен қатар қызметкерлерді бағалаудың объективтілігі мен сапасын арттыруға арналған. Сонымен қатар, бұл ұйым директоры үшін әзірленіп жатқан жобалардың барысын бақылаудың ыңғайлы құралы.

Аннотация

«Информационная система для проектно-строительной организации» разработана с целью автоматизации процесса управления проектированием и работой проектной группы организации.

Данная система реализована в виде одностраничного кроссплатформенного адаптивного веб-приложения. В ходе анализа современных методов управления проектированием были выбраны наиболее подходящие методы – Kanban и Scrum. На основе этих подходов к управлению была создана собственная методология управления проектами, которая отвечает индивидуальным требованиям организации.

Анализ аналогичных систем, основанных на Kanban-методе, показал, что современные системы управления проектами имеют весьма нагруженный и неинтуитивный интерфейс, не отвечают индивидуальным требованиям организации, имеют избыточный функционал, а также для изучения некоторых веб-сервисов для управления проектами требуется время. В связи с чем, наиболее оптимальным выходом для организации стало создание собственного веб-приложения.

Основным требованием к системе стал простой и интуитивно понятный интерфейс, который будет удобен для разных возрастных групп и для сотрудников с различной компьютерной грамотностью.

Результат, проведенной в ходе написания дипломного проекта, работы – кроссплатформенное и адаптивное веб-приложение, позволяющее осуществлять контроль над работой проектной группы, а повышающее качество и скорость работы сотрудников, за счет визуализации приоритетных задач и правильного тайм-менеджмента.

Разработанное приложение призвано повысить эффективность работы организации, а также повысить объективность и качество оценки работы сотрудников. Помимо этого, это удобный инструмент для отслеживания прогресса по разрабатываемым проектам для руководства организации.

Annotation

The «Information System for Design and Construction Organization» was developed to automate the process of managing designing and work of the organization's project team.

This system is implemented on the base of Single Page cross-platform and adaptive Application. During the analysis of modern project management methods, it turned out that the most appropriate methods for project management in organizations are Kanban and Scrum. Based on this management methods own methodology was created that meets the individual requirements of the organization.

The analysis of similar systems based on the Kanban method showed that modern project management systems have a very loaded and non-intuitive interface, do not meet the individual requirements of the organization, have excessive functionality, and it takes time to study some web services for project management. Therefore, the most optimal solution for the organization was the creation of its own web application.

The main requirement for the system was a simple and intuitive interface that would be convenient for different age groups and for employees with different computer skills.

The result of writing this thesis is a cross-platform and adaptive Single Page Application that allows you to control the work of the project team and improves the quality and speed of employees by visualizing priority tasks and proper time management.

The developed application is designed to increase the efficiency of the organization, as well as to increase the objectivity and quality of the assessment of employees. Besides, it is a convenient tool for tracking the progress of projects under development for the director of the organization.

Содержание

Введение	8
1 Аналитическая часть	10
1.1 Анализ методологий и методов управления проектами	10
1.2 Анализ современных систем управления проектами	15
1.3 Понятие информационной системы и классификация ИС	20
1.4 Понятие веб-приложения и его виды	21
1.5 Описание предметной области	22
1.6 Средства разработки веб-приложения	23
1.7 Вывод	27
2. Разработка информационного обеспечения системы	28
2.1 Постановка задачи	28
2.2 Проектирование функциональной структуры ИС	29
2.3 Проектирование бизнес-модели и оптимизации бизнес-процессов	30
2.4 Проектирование информационного обеспечения	35
3 Разработка программного обеспечения системы	48
3.1 Разработка веб-приложения	48
3.2 Контрольный пример	56
4 Безопасность жизнедеятельности	63
4.1 Анализ условий труда при разработке ИС	63
4.2 Расчет аспирационной системы для производственного помещения	65
4.3 Эргономические требования к рабочему месту в офисе	70
4.4 Вывод	73
5. Экономическая часть	74
5.1 Резюме	74
5.2 Расчет трудоемкости разработки ИС	74
5.3 Расчет затрат на разработку ИС	78
5.4 Расчет эксплуатационных затрат	82
5.5 Расчет результатов от создания и использования ИС	84
5.6 Расчет основных показателей экономической эффективности	86
Заключение	88
Список литературы	90
Приложение А Техническое задание	93
Приложение Б Листинг программы	94
Приложение В Акт внедрения	100

Введение

В настоящее время умение управлять проектом является одним из важнейших факторов при проектировании, наравне с постановкой задачи, определением требований и непосредственным проектированием. Грамотное управление проектом позволит осуществить быструю и эффективную разработку этого проекта. Такое управление может повысить продуктивность участников проекта за счет правильного распределения выполняемых задач, сроков и приоритетов.

Для реализации проекта необходимо выполнить какие-то условия, цепочку действий. Казалось бы, что тут может быть сложного – можно составить список всех задач и выполнять ее за другой. Такой подход имеет место быть, к примеру в походе за продуктами. Хотя и там список продуктов, написанный в случайном порядке, не принесет желаемых результатов, если есть ограничения по количеству пакетов, которые можно унести за раз, по деньгам и конечно же по времени. Именно для этого и нужно управление, которое даже в случае с походом в магазин, показывает свою эффективность.

Каждый проект уникален и может быть стандартизован лишь относительно. Управление проектами – это своеобразная последовательности действий и область деятельности, которая позволяет определить и достичь определенных целей, поставленных по проекту и балансировать между объёмами выполняемых работ, ресурсами, временем, качеством и рисками.

Основопологающим фактором успеха эффективного управления проектами является то, что такое управление предполагает грамотное планирование и реализацию проекта, за счет оптимизации временных, денежных затрат и человеческих ресурсов, но при этом запланированное качество конечного продукта не страдает.

Существует большое количество методологий по управлению проектами. Основной их целью является формализация процессов в проектировании, которая реализуется посредством стандартизации процессов управления. Эти методологии различаются по областям применения, детализированности, самодостаточности и формализации. Самые популярные на сегодняшний день методы и методологии: традиционный (классический) проектный подход, agile, PRINCE2, SCRUM, Kanban и т.д.

В условиях современного мира возникает необходимость создания программного обеспечения для управления проектами. Использование таких систем позволяет организации реализовать прозрачность и оценить затраты (финансовые, временные, человеческие ресурсы) в данном проекте. На данный момент функционируют десктопные, веб- и мобильные приложения. Примерами могут служить сервисы bitrix24, ganttpro, системы ELMA, Advanta. Но большинство таких систем бывают сложны, имеют высокую стоимость подписки, имеют много лишнего или не нужного функционала, не могут подстроиться под специфику работы компании. К тому же, большое количество организаций стремятся иметь собственные программные

продукты, которые будут отвечать их индивидуальным требованиям и при необходимости могут быть доработаны.

Еще один выбор, который стоит перед компанией – в каком формате такого программного обеспечения по управлению проектами будет наиболее удобно работать сотрудникам. Ведь не у всех сотрудников одинаковые операционные системы, не у сотрудника будет доступ к компьютеру, а следовательно, и к десктопному приложению. Конечно же можно реализовать мобильную и десктопную версии приложения. Но в настоящее время развивается новое и во многом наиболее удобное направление – создание веб-приложений. Это могут быть как одностраничные веб-приложения, адаптированные под разные экраны, так и прогрессивные веб-приложения, которые подобно мобильным приложениям можно установить на главный экран смартфона, вне зависимости от операционной системы этого смартфона.

Исходя из вышесказанного, цель дипломной работы – разработка информационной системы для управления проектами на базе адаптивного веб-приложения. Основными преимуществами системы должны стать ориентированность на индивидуальные требования организации, простой и интуитивно понятный пользовательский интерфейс, адаптивность и кроссплатформенность.

Для достижения цели необходимо выполнить следующие задачи:

- изучить предметную область;
- проанализировать аналогичные системы;
- определить требования к приложению;
- осуществить непосредственную разработку приложения;
- произвести тестирование и отладку приложения;
- внедрить результат разработки в производство.

Объект исследования дипломной работы – проектная группа организации «ЭлитСтройПроект-КС» и ее работа, которая заключается в проектировании различных строительных объектов. Предметом исследования служит процесс управления работой проектной группы. Улучшение этого процесса за счет использования современных технологий и методологий позволит повысить эффективность работы проектной группы, сбалансировать денежные, временные затраты, а также человеческие ресурсы.

1 Аналитическая часть

1.1 Анализ методологий и методов управления проектами

За всю историю человечество успело реализовать огромное количество различных проектов, будь то отправка человека в Космос или строительство пирамид. Но любой из этих проектов не был бы так успешен без правильного управления.

Управление проектом – это организация всех необходимых моментов для достижения цели, контроль работы участников проекта. Правильный менеджмент проекта позволяет добиться успеха.

Каждый проект уникален. Условия, в которых разрабатывается проект всегда разные, люди, принимающие участие в проектах, отдельные личности со своими талантами, навыками и умениями. Именно поэтому не существует идеальной системы управления проектами, которая бы подходила всем проектам. Нет и методологии, которая бы подходила всем руководителям и их сотрудникам. Тем не менее на данный момент существует большое количество методологий управления проектами, которые могут быть применены в управлении проектированием и могут направить управление в нужное русло[1].

Самые распространенные подходы к управлению проектами:

а) классический проектный менеджмент или традиционная (каскадная) методология управления проектами представляет собой поток последовательных этап проектного управления: определение требований, проектирование, реализация, внедрение, тестирование и отладка, установка, эксплуатация и сопровождение. В такой методологии переход от одного этапа к другому возможен, только в том случае, если предыдущий этап полностью завершен и не требует доработок. Проблема такой методологии в том, что она не гибкая и не может подстраиваться под все изменения пожеланий клиента [2]. Схема подхода изображена на рисунке 1.1;

б) Agile – это семейство гибких подходов к управлению проектами. В соответствии с этим подходом, проект разбивается на маленькие подпроекты, которые потом собираются в готовый проект. Основные идеи такого подхода:

- 1) люди и их взаимодействия важнее процессов и инструментов;
- 2) работающий продукт важнее исчерпывающей документации;
- 3) сотрудничество с заказчиком важнее согласования условий контракта;
- 4) готовность к изменениям важнее следования первоначальному плану[3].

Схема подхода представлена на рисунке 1.2.

Одно из важнейших достоинств данного подхода – это его умение подстроиться под любые условия и процессы организации. Также сильной стороной является быстрое и безболезненное реагирование на изменения.

Такой подход значительно упростит управление проектов с «открытым концом».



Рисунок 1.1 – Схема классического проектного управления

Слабые стороны состоят в том, что такой подход более абстрактный и систему управления организации нужно будет создавать свою, на основе основных принципов Agile.



Рисунок 1.2 – Схема работы по Agile

в) Scrum – гибкий подход, разработанный на основе Agile. Сам термин пришел из игры регби, где это слово scrum означает схватку. Принцип подхода заключается в разбиении работы по проекту на жестко фиксированные по времени итерации, которые предоставляют заказчику какую-либо готовую часть проекта. Такие итерации называются спринтами.

Перед началом каждого спринта проводится оценка содержимого проекта на соответствие требованиям заказчика[6].

Такой метод управления подойдет для проектов, в которых важны быстрые результаты и постоянная готовность к изменениям. Недостатком такого подхода считается его требовательность к команде проекта. Члены команды должны обладать множеством умений, даже вне своей компетенции, а также должны уметь работать в команде и активно брать на себя ответственность.

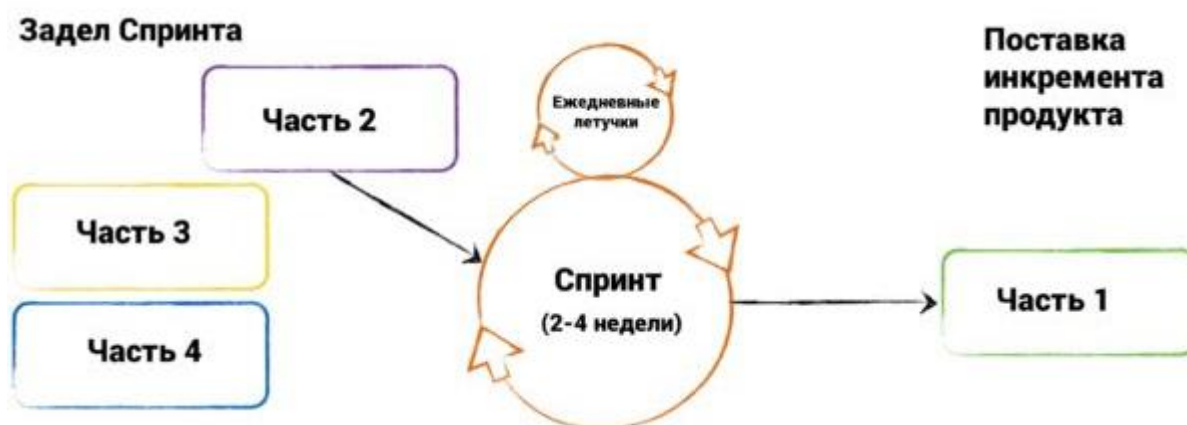


Рисунок 1.3 – Схема процесса проектирования по Scrum

г) Lean похож на Scrum в разбиении проекта на небольшие пакеты работ, но в отличие от последнего не дает четких указаний по тому, как управлять разработкой такого пакета. Как и в классическом управлении эти пакеты могут быть этапами определения требований, проектирования, реализации. Однако в отличие от классического метода, Lean позволяет выполнять несколько задач на разных этапах. За счет этого повышается гибкость и скорость исполнения проектов[1].

Схема работы Lean показана на рисунке 1.4.

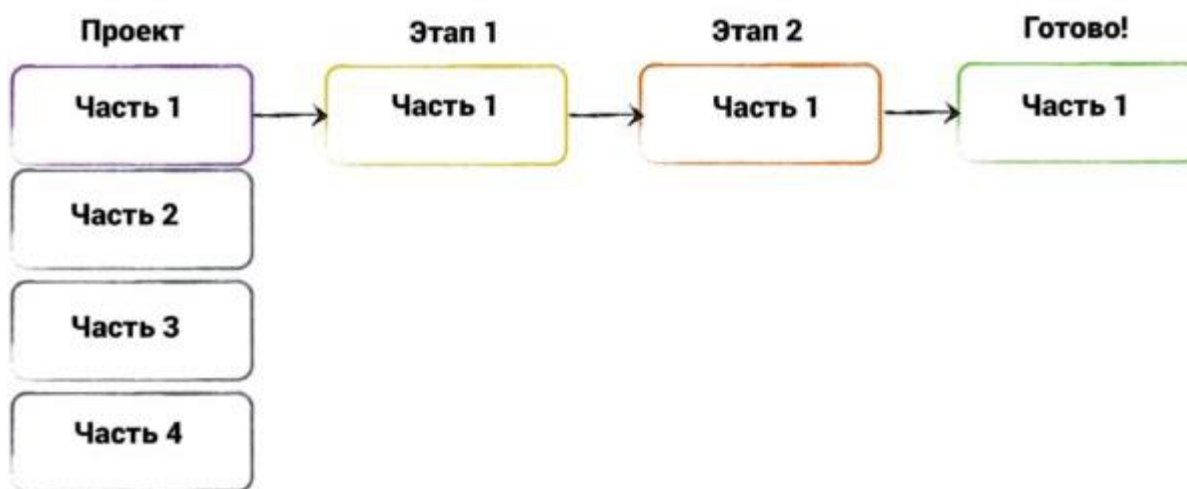


Рисунок 1.4 – Схема работы по Lean

д) Kanban был изобретен инженером компании Toyota и представляет собой процесс схожий с производством детали. Суть такова: на вход процесса

поступает кусочек металла, а на выходе получается деталь. Также в Kanban используется принцип супермаркетов: «Держи на полке то, что необходимо клиенту». Смысл состоит в том, что задачи выполняются по приоритету, т.о. те задачи, которые не требуют срочного выполнения, могут остаться неоконченными и останутся на своем этапе, до тех пор, пока не выполнятся задачи с более высоким приоритетом. Схема похода отображена на рисунке 1.5.



Рисунок 1.5 – Схема работы по Kanban

Метод Kanban не такой строгий как Scrum, и в отличие от последнего один член команды может выполнять несколько задач. Удобство Kanban состоит в том, что этот метод отлично визуализирован. Весь принцип этого метода в том, чтобы перетаскивать задачи в зависимости от стадии, в которой находится задача[5].

Такой метод универсален и применим почти к каждому проекту. При правильном распределении задач между участниками проекта, а также расстановке ограничений можно сэкономить ресурсы и укладываться в сроки.

е) 6 сигм – это структурированная версия Lean, а также этот метод во многом похож на Kanban. Состоит из пяти шагов (DMEDC):

- 1) Define – определение содержание проекта, его предпосылок, постановка целей;
- 2) Measure – измерение: анализ количественных данных о проекте, показателях, определяющих его успех;
- 3) Explore – исследование: поиск способов достижения цели;
- 4) Develop – разработка: реализация планов и решений, принятых на предыдущих этапах, на этом этапе производится измерение прогресса проекта;
- 5) Control – контроль: на данном этапе документируются полученные знания и опыт для улучшения процессов реализации проектов в дальнейшем.

Схема работы 6 сигм представлена на рисунке 1.6.

ж) PRINCE2 – PProjects IN Controlled Environments version 2, что означает «Проекты в контролируемой среде версия 2». Этот подход

основывается на классическом проектном управлении и 6 сигмах. Базируется на 7 принципах, 7 процессах и 7 темах проекта. Схема подхода отображена на рисунке 1.7.[4].

Основными преимуществами этого метода можно считать хорошую адаптируемость к требованиям организации, четкое распределение ответственности, так же, как и в методе 6 сигм – акцентирование на продуктах проекта. Фокус в данном подходе направлен на экономическую целесообразность проекта. От метода Kanban в эту методологию пришел кайзен – акцент на постоянном совершенствовании.



Рисунок 1.6 – Схема метода 6 сигм

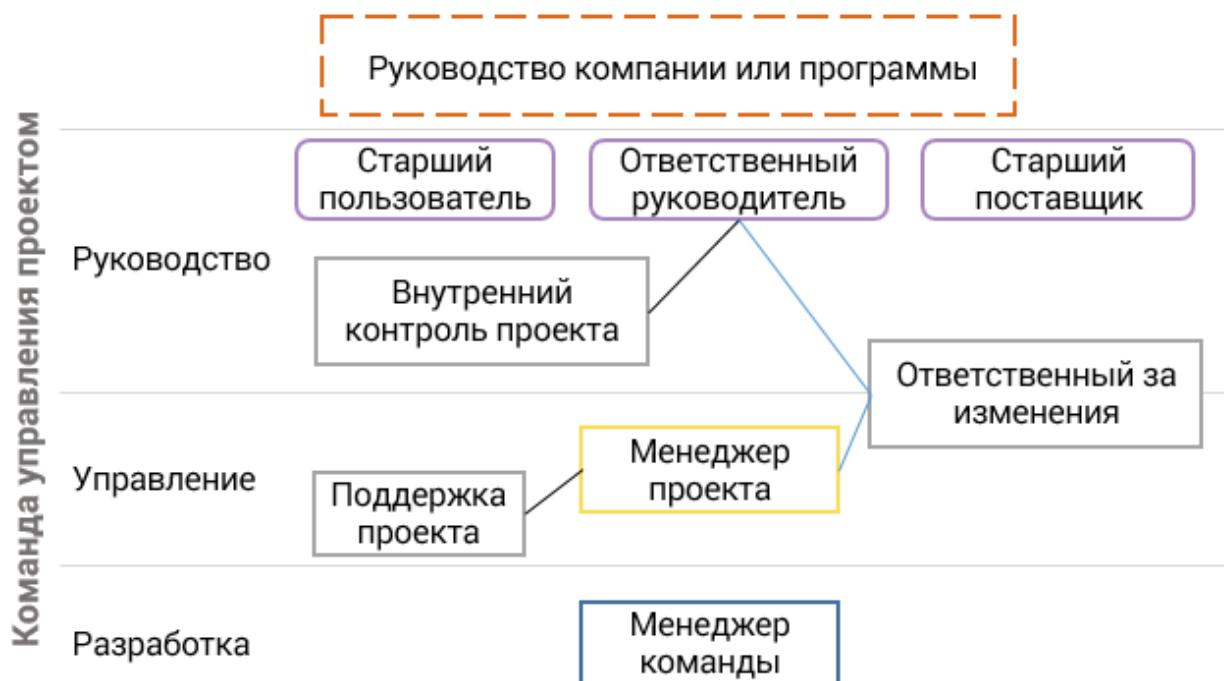


Рисунок 1.7 – Схема ролей в PRINCE2

Вывод. В ходе анализа популярных методов управления, было принято решение создать собственную систему управления на основе Kanban и Scrum.

Именно сочетание этих методов и использование только их лучших сторон позволит оптимизировать и автоматизировать процесс управления работой проектной группы в организации «ЭлитСтройПроект-КС».

1.2 Анализ современных систем управления проектами

Системы управления проектами можно рассматривать в различных смыслах. В широком понимании – это совокупность технологических ресурсов и организационных методов, осуществляющих поддержку проекта, направленные на повышение эффективности реализации этого проекта. Система управления проектами с точки зрения используемых средств управления – это такая автоматизированная информационная система, или проще говоря программа, которая позволяет организовать процесс управления проектированием. На данный момент существует большое количество таких систем, созданных под определенные методики, области деятельности, в разных форматах. В анализе будут рассмотрены наиболее популярные системы управления проектами, реализованные в виде веб-приложений.

а) GanttPRO – современная система управления проектами в виде диаграммы Ганта. Задачи в этой диаграмме размещаются по вертикали, а их сроки выполнения – по горизонтали. В результате формируется график, позволяющий понять объем работ и сроки выполнения. Помимо диаграммы Ганта, GanttPRO позволяет управлять ресурсами, стоимостью проекта и командной работой[7].

Имеются готовые шаблоны. На рисунке 1.8 представлена доска задач в GanttPRO.

Основные преимущества:

- кроссплатформенность;
- интуитивный интерфейс;
- быстрое изучение инструментов для работы;
- подходит для управления как личными, так и командными проектами;
- возможность переключения на Kanban доску – доску задач;
- готовые шаблоны.

Недостатки:

- невозможность отображения всех проектов на одном экране;
- отсутствие тайм-трекинга;
- малое количество интеграций.

б) Bitrix24 – система управления проектами, имеющая большое количество возможностей (CRM, диаграмма Ганта, Kanban и т.д.)

Система управления взаимоотношениями с клиентами (CRM, CRM-система, сокращение от англ. Customer Relationship Management) — прикладное программное обеспечение для организаций, предназначенное для автоматизации стратегий взаимодействия с заказчиками (клиентами), в частности для повышения уровня продаж, оптимизации маркетинга и улучшения обслуживания клиентов путём сохранения информации о клиентах

и истории взаимоотношений с ними, установления и улучшения бизнес-процессов и последующего анализа результатов[7].

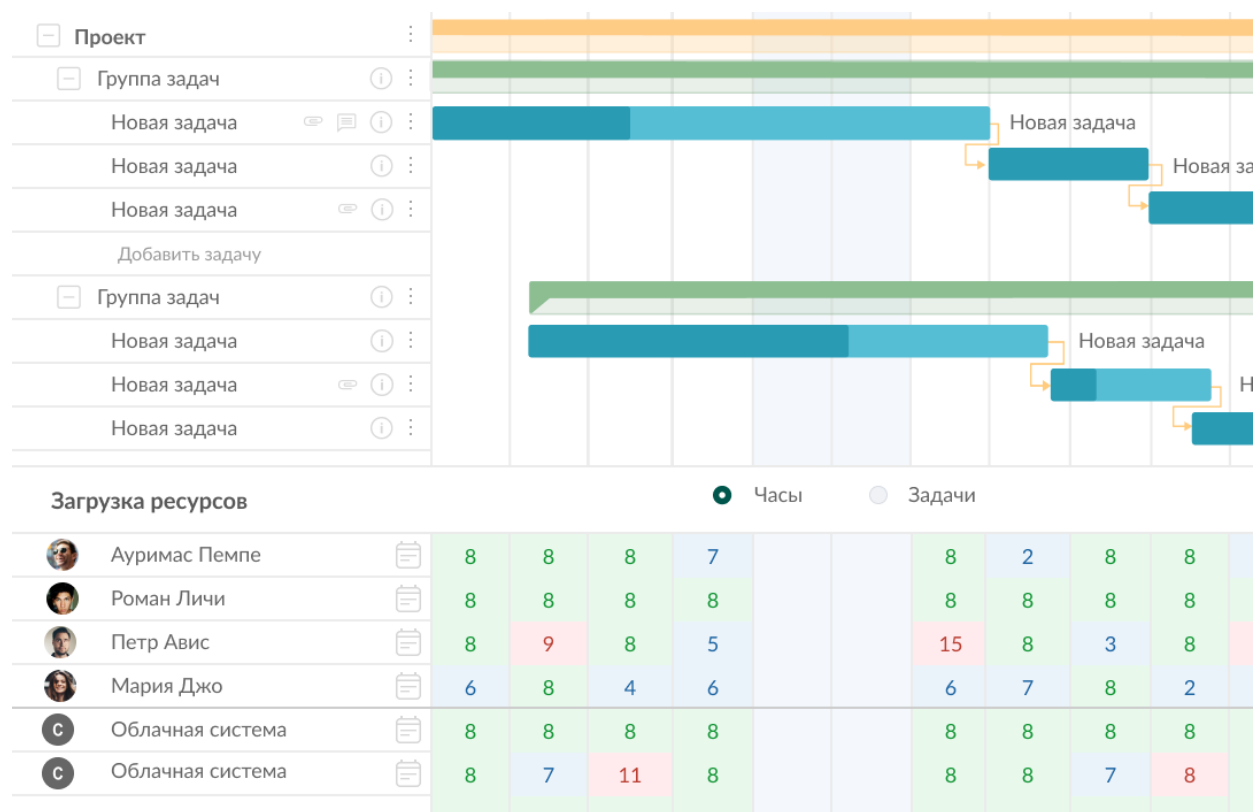


Рисунок 1.8 – Диаграмма задач в сервисе GanttPRO

CRM используется для продаж, а контакт-центр объединяет каналы коммуникации с клиентами. Имеется возможность создания сайтов и интернет-магазинов, назначения ролей, уведомления.

Преимущества:

- создание шаблонов для повторяющихся проектов;
- приложения для IOS и Android;
- связывание задач с системой управления взаимоотношениями с клиентами;
- возможность адекватной и объективной оценки эффективности работы сотрудников.

Недостатки:

- неудобный и не интуитивный интерфейс;
- требуется время на изучение инструментов, обучение и внедрение.
- в большей степени подходит для работы с CRM;
- бесплатный пробный период действует неограниченное время, только для маленьких команд до 12 человек, т.о. подходит только для маленьких организаций.

На рисунке 1.9 изображена рабочая область участника проекта в Bitrix24.

в) Wrike – система управления проектами, ориентированная на большие команды. Есть возможность распределения обязанностей и приоритетов, управления ресурсами, загрузкой сотрудников, обновления статуса работы. Проще говоря, система имеет большое количество возможностей для повышения эффективности работы команды. На рисунке 1.10 представлен веб-сервис Wrike.

Преимущества:

- мобильные приложения для IOS, Android;
- постоянная синхронизация;
- возможность составления интерактивных отчетов.

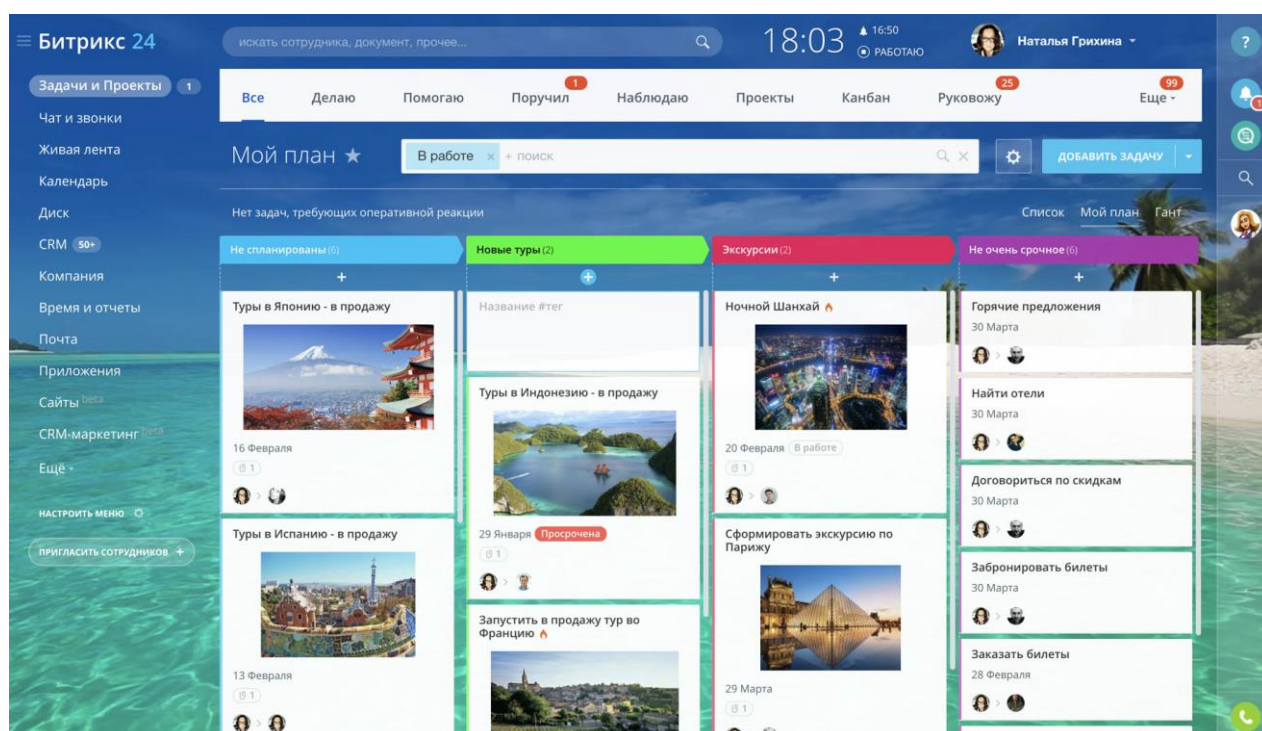


Рисунок 1.9 – Канбан-доска в Bitrix24

Недостатки:

- не интуитивный интерфейс;
- высокая стоимость;
- долгий период ознакомления и изучения инструментов для работы.

г) Asana – система управления проектами для небольших команд и для личного пользования. Asana позволяет фокусироваться на целях, проектах, а также ежедневных задачах и дедлайнах.

Подходит для ведения маркетинговых и креативных кампаний, проектов в дизайне, IT отделах, HR, финансах, продаже, разработке ПО, организации мероприятий.

Преимущества:

- красивый дизайн;
- большое количество синхронизаций;
- большое количество сфер применения.

Недостатки:

- не предназначен для больших команд;
- не поддерживается русский язык;
- отсутствие полноценной диаграммы Ганта.

На рисунке 1.11 изображена доска задач в веб-сервисе Asana.

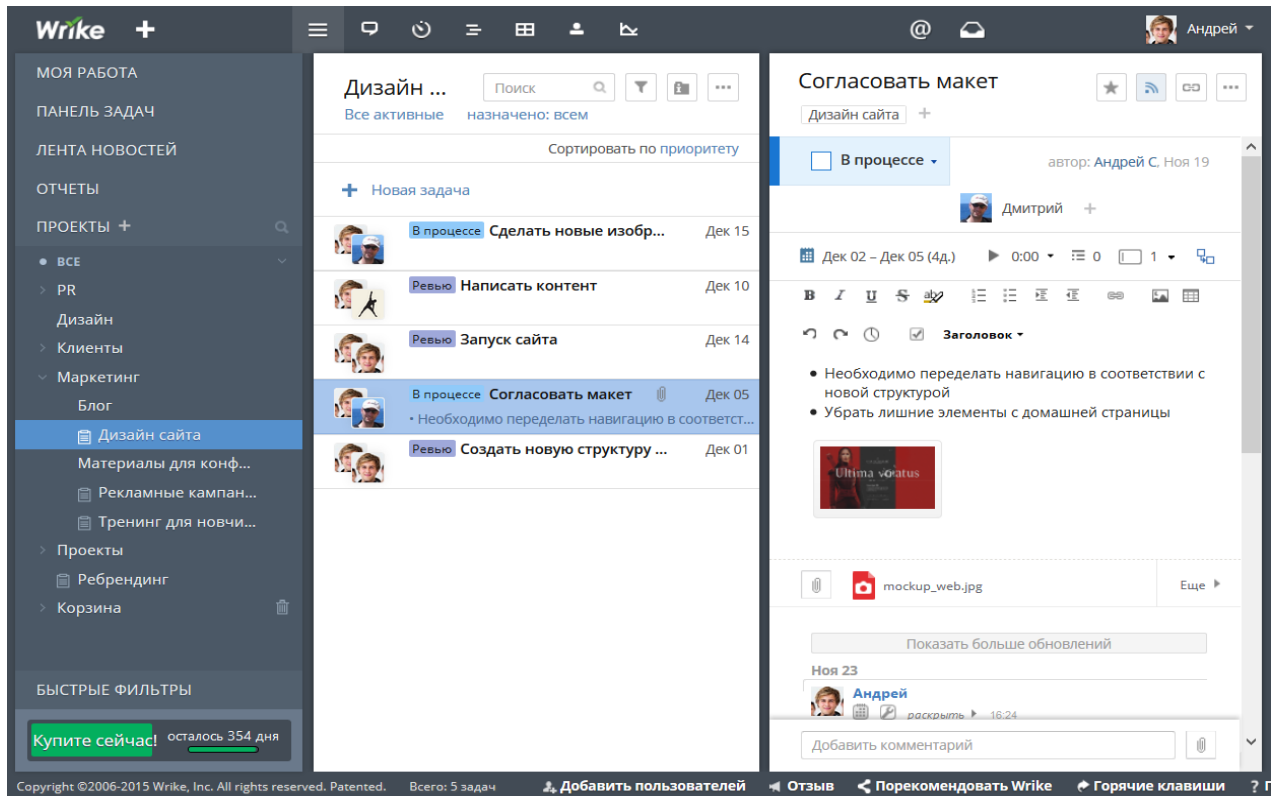


Рисунок 1.10 – Веб сервис Wrike

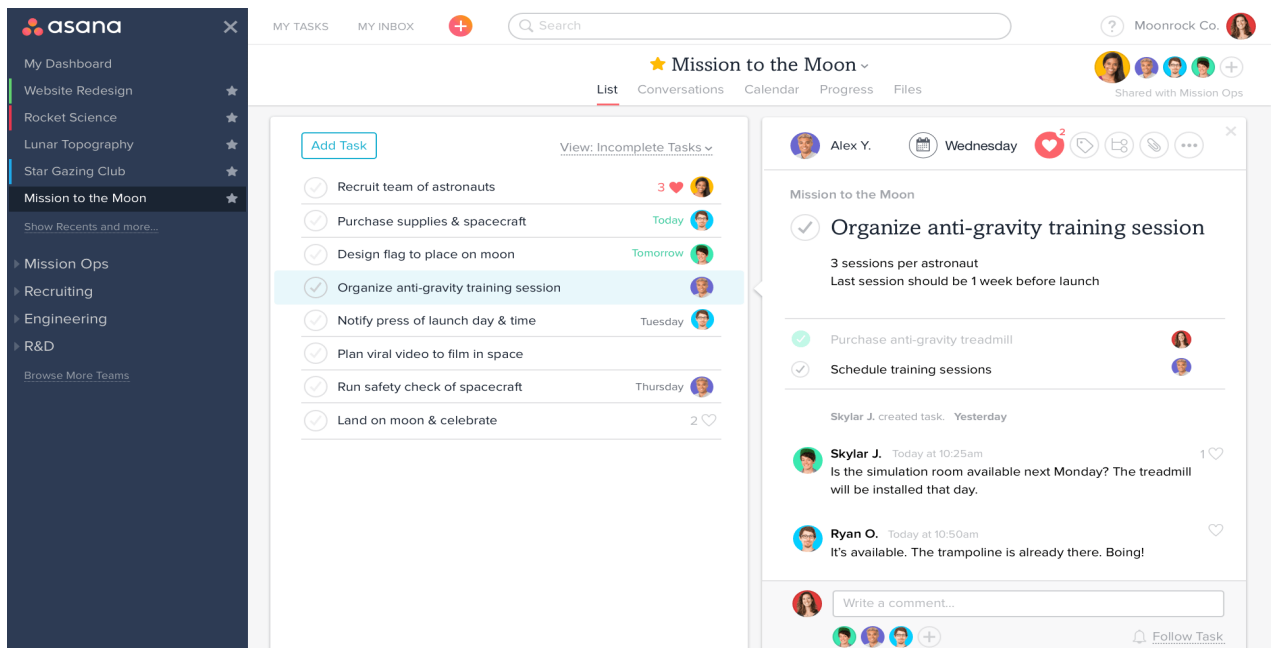


Рисунок 1.11 – Веб-сервис Asana

д) Basecamp – функциональная система для управления. Все собрано в одном месте и все члены команды будут знать, что им нужно делать и в каком состоянии находится проект. Состоит из нескольких пространств: списки дел, доски сообщений (коммуникации по определенным темам), графики, хранилище (все документы, файлы находятся в одном месте), групповой чат, вопросы (заменять собрания и совещания)[7].

На рисунке 1.12 отображено рабочая область для дизайнеров, содержащая документы и файлы по проекту, в сервисе Basecamp.

Преимущества:

- наличие множества пространств;
- возможность составления отчетов;
- возможности поиска: по ключевым словам, людям, везде или в отдельном пространстве, по изображениям и файлам;

1) публичные ссылки для третьих лиц вне проекта.

Недостатки:

- требуется время на изучение.
- не поддерживается русский язык.
- недостаточно интеграций.
- со временем проект накапливает множество деталей, в которых трудно разобраться.

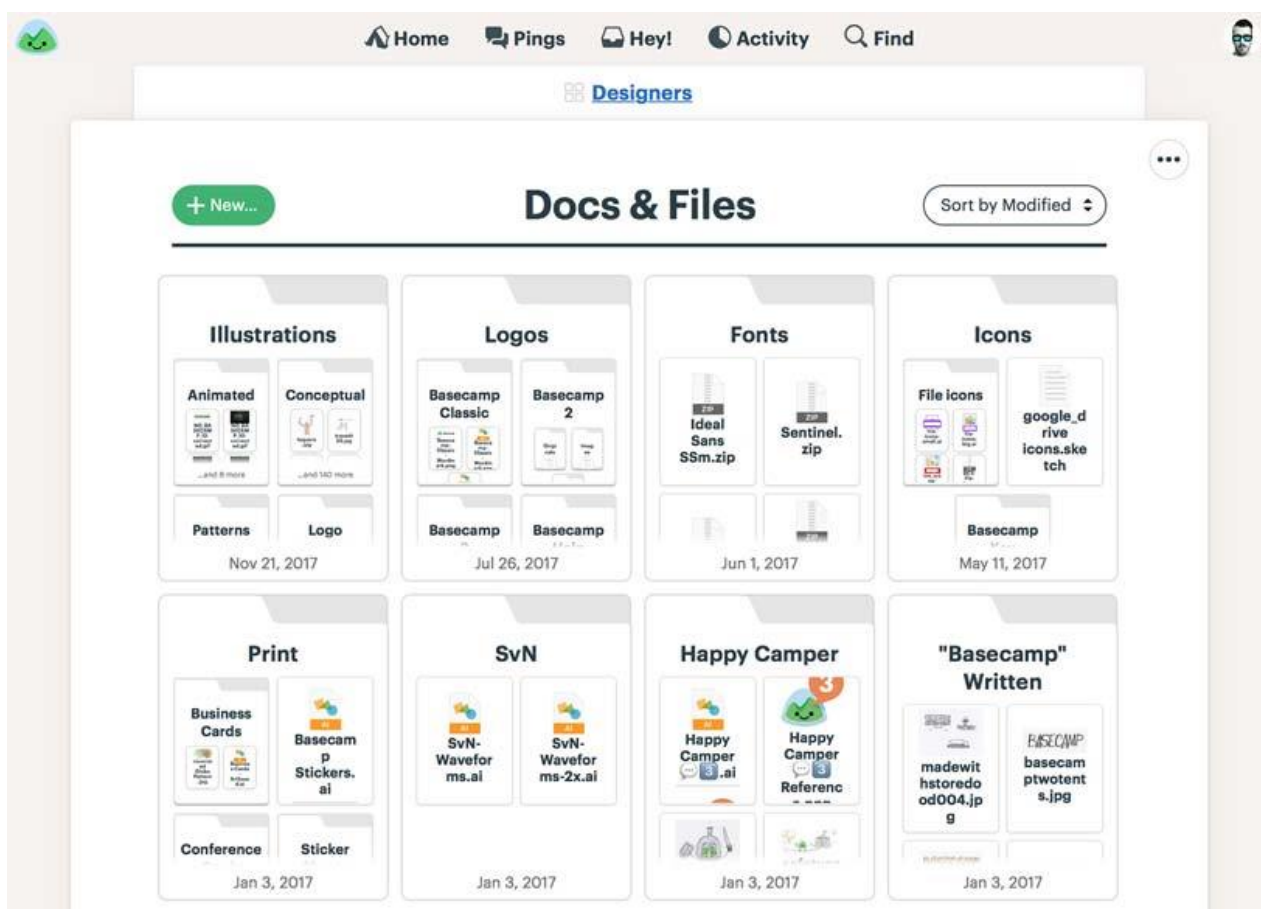


Рисунок 1.12- Веб-сервис Basecamp

Вывод. Данный анализ позволил выявить основные преимущества и недостатки современных систем управления проектами. Что установило, какими моментами данных сервисов можно вдохновиться и применить похожее у себя и на какие ошибки стоит обратить внимание.

1.3 Понятие информационной системы и классификация ИС

Информационная система – это система, предназначенная для сбора, хранения, поиска и обработки информации, а также соответствующие человеческие, технические и финансовые ресурсы, обеспечивающие и распространяющие эту информацию [10].

Существует несколько классификаций информационных систем: по архитектуре, по степени автоматизации, по характеру обработки данных, по сфере применения, по масштабности [11].

Классификация по архитектуре изображена на рисунке 1.13.

В настольных (или локальных ИС) все компоненты (СУБД, приложения) расположены на одном компьютере, а в распределенных соответственно на разных. В файл-серверных распределенных ИС база данных расположена на файловом сервере, а система управления базой данных и большинство клиентских приложений на так называемых рабочих станциях. Что касается клиент-серверных распределенных ИС, то у них БД и СУБД находятся на сервере, а клиентские приложения на рабочих станциях.



Рисунок 1.13 – Классификация ИС по архитектуре

Двухзвенные клиентские ИС имеют два звена – сервер БД (там же и располагаются БД и СУБД) и рабочие станции с клиентскими приложениями).

В многозвенных ИС подключаются серверы приложений – которые позволяют клиентском ПО не взаимодействовать напрямую с СУБД. К таким ИС относятся все современных веб-приложения, которые помимо звена СУБД и клиентской части (выполняющейся в браузере) имеют также веб-сервер.

Классификация по степени автоматизации, сфере применения, по характеру информации изображена на рисунке 1.14.

В большинстве случаев, когда идет разговор об информационных системах, подразумеваются именно автоматизированные информационные системы. Это системы с неполной автоматизацией, требующей вмешательства человека. И хоть во все ныне действующие классификации включены ручные ИС, в настоящее время не совсем корректно использовать этот термин, поскольку все определения информационных систем предписывают наличие компьютера в составе ИС.

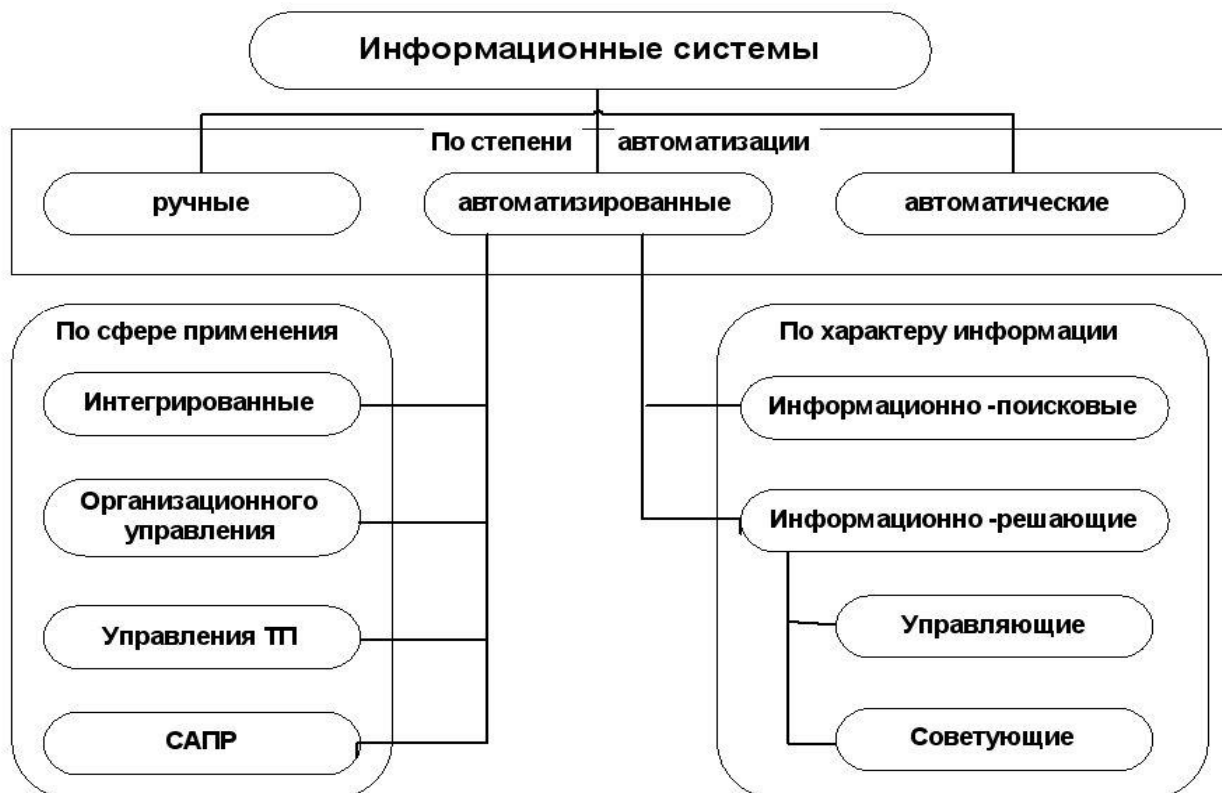


Рисунок 1.14 – Классификация ИС

1.4 Понятие веб-приложения и его виды

Веб-приложение – это приложение, использующее клиент-серверную архитектуру, в котором взаимодействие между клиентом и веб-сервером происходит посредством браузера. Хранение информации производится на сервере, а сам обмен информацией происходит по сети. Преимущество веб-приложений состоит в их кроссплатформенности, т.е. они не зависят от операционной системы.

Веб-приложение состоит из back-end – серверной части и front-end – клиентская часть, выполняемая в браузере.

В настоящее время стали набирать популярность одностраничные приложения (SPA – Single Page Application), а также прогрессивные веб-приложения (PWA – Progressive Web Application).

SPA – одностраничное веб-приложение, которое загружается на одной HTML-странице и не перезагружается, за счет динамического обновления. Такое приложение выполняется внутри браузера, динамическое обновление осуществляется с помощью AJAX, а отрисовка кода производится на стороне пользователя. Вместо перезагрузки страницы в SPA приложениях подгружаются отдельные компоненты. По сравнению с обычными веб-сайтами SPA приложения работают быстрее, а разработка таких приложений обычно направлена на адаптивность под все устройства. Примерами SPA могут служить Gmail, Vk, Facebook и т.д.

PWA – простыми словами – веб-сайт, который ведет себя как приложение. Его можно добавлять на экран мобильных устройств, он может отправлять push-уведомления. PWA может подключиться к аппаратным средствам устройства и работать без подключения к интернету. Так же, как и одностраничные приложения PWA обладает высокой скоростью по сравнению не только с традиционными сайтами, но и нативными приложениями. Основное отличие PWA от SPA в том, что PWA может работать в оффлайн-режиме и устанавливаться на само устройство, вне зависимости от его операционной системы. В качестве примера PWA может выступить сервис Google Docs, который можно использовать в оффлайн-режиме[8].

Принцип работы PWA-сайтов состоит в том, что при первом посещении такого сайта устанавливается скрипт Service Worker, добавляющие оболочку приложения в кэш. Далее после загрузки оболочки, приложение сначала запрашивает содержимое для представления, а потом и контент. Когда все эти запросы завершены Service Worker отправляется в режим ожидания до следующих сетевых запросов.

SPA может быть PWA, но PWA не всегда может быть SPA. Говоря простым языком, PWA это изменение существующего сайта, будь то многостраничное или одностраничное приложение. Поэтому на начальных этапах возможна разработка SPA, а в дальнейшем его перевод в PWA.

Вывод. На данном этапе для автоматизации процесса управления проектированием может использоваться одностраничное приложение (SPA) с динамическим подгрузением компонентов.

1.5 Описание предметной области

ОО «ЭлитСтройПроект-КС» позиционирует себя как одного из лидеров строительного рынка страны. Компания была создана 1999 году. За этот период она прошла путь: от компании, стоявшей у истоков строительного бизнеса и занимавшейся строительством отдельных домов, до известного участника строительного рынка, возводящего и застраивающего целые кварталы протяженностью на десятки гектаров.

В настоящее время организация специализируется на возведении офисных помещений класса “А”, жилых комплексов класса «элит», а с 2005 г. компанией было принято решение развивать проекты в классах «бизнес» и «комфорт». При этом, группа компаний осуществляет полный цикл строительства: от этапа проектирования архитектурного замысла, проведения строительно-монтажных работ, до ввода объекта в эксплуатацию и реализации клиентам.

Проекты компании: Бизнес-центр Кулан, Офисное здание KCELL, Эталонный центр, Мечеть Астана, LUXOR, Жилой комплекс Тау Самал и т.д.

Организационная структура проектно-строительной организации выглядит следующим образом см. рисунок 1.15.

Область, подлежащая исследованию в данной работе – работа проектной группы организации.

Проектная группа состоит из куратора, специалистов и главного инженера проекта (ГИП). При разработке проекта, учитываются сроки начала и окончания проектирования, определяются задачи для выполнения и назначаются специалисты. Специалисты выполняют соответствующие своей профессиональной области задачи, которые определяются ГИПом, а назначаются куратором. Директор проектной организации может назначать ГИПов и кураторов на проекты.

Управление проектами в данной организации осуществляется посредством электронной почты, в печатной, либо в устной форме. То есть задачи, необходимые для выполнения в ходе разработки проекта, сообщаются лично, посредством электронных писем, либо напечатанные на бумаге. Это создает определенные трудности, так как определенное количество работников находятся на удаленной работе, для связи с ними приходится использовать электронную почту. Все файлы и документы по проектам хранятся на сетевом диске, что также создает некоторые трудности, для сотрудников, не находящихся в офисе, и не позволяет руководству оценивать прогресс по проекту.

Таким образом, в рамках разрабатываемой системы, можно выделить сущности, подлежащие проектированию в проектной части: проект, задача по проекту, сотрудник – пользователь системы, руководитель проекта, который назначает сотрудникам задачи.

1.6 Средства разработки веб-приложения

Для разработки веб-приложения по управлению работой проектной группы выбрана связка фреймворков – Laravel+Vuejs. Laravel является фреймворком скриптового языка PHP, а Vuejs в свою очередь фреймворк JavaScript.

Фреймворк – это такое программное обеспечение, облегчающее разработку компонентов проекта, которое, в отличие от библиотек, накладывает на архитектуру приложения ограничения. Таким образом

фреймворк можно считать так называемым каркасом приложения, который расширяется и изменяется согласно заранее определенным критериям.

Laravel является PHP-фреймворком для веб-приложений. Он облегчает работу с такими задачами как аутентификация, маршрутизация, пагинация, кэширование и т.д. Основными преимуществами Laravel можно считать:

- понятная документация;
- мощная экосистема;
- гибкая система маршрутизации;
- удобный механизм обработки ошибок и исключений;
- встроенные ООП библиотеки;
- встроенные и достаточно хорошие механизмы аутентификации и авторизации пользователей, а также механизмы кэширования.

Именно эти отличительные черты данного фреймворка определили его популярность в сравнении с другими PHP-фреймворками (yii, Codeigniter, Zend, CakePHP). Статистика за последний год представлена на рисунке 1.16.

Vue фреймворк ЯП JavaScript предназначенный для создания пользовательских интерфейсов, выступает также как веб-фреймворк для разработки одностраничных приложений. К основным преимуществам данного фреймворка можно отнести:

- маленький размер фреймворка – всего 18-21 КБ, т.е. для его загрузки не требуется много времени;
- легок для изучения и разработки – имеет простую структуру, можно разрабатывать и большие и малые шаблоны, что значительно экономит время;
- простая интеграция – Vuejs основан на JavaScript, соответственно может быть интегрирован в другие приложения, построенные на JS, может использоваться как для разработки новых приложений, так и для изменения уже существующих;
- подробная и понятная документация;
- гибкость – за счет виртуальных узлов пользователь может записать шаблон в HTML-файл, JS-файл и файл чистого JavaScript, такая гибкость облегчает понимание для разработчиков, использующих React или Angular;
- двусторонняя связь, которая обеспечивается за счет архитектуры MVVM, позволяющая обрабатывать блоки HTML.

Vue вобрал в себя все самое лучшее от фреймворков созданных ранее – React, Angular. Так же, как и React, Vue использует Virtual DOM, что обеспечивает быструю и почти безошибочную производительность. И хотя в сравнении со своими предшественниками данный фреймворк еще не так популярен, ожидается, что ближайшее время он будет набирать все большую и большую популярность.

К крупнейшим проектам, использующим Vue, относятся такие популярные ресурсы как: 9Gag – социальная сеть, позволяющая делиться трендовым контентом, а также GitLab – менеджер репозитория, предназначенный главным образом для разработки и дублирования кода.

И если Angular и React поддерживаются крупнейшими компаниями, такими как Facebook и Google, то Vue популярен в сообществе открытого кода.

ОРГАНИЗАЦИОННАЯ СТРУКТУРА ТОО «ЭЛИТСТРОЙПРОЕКТ-КС»

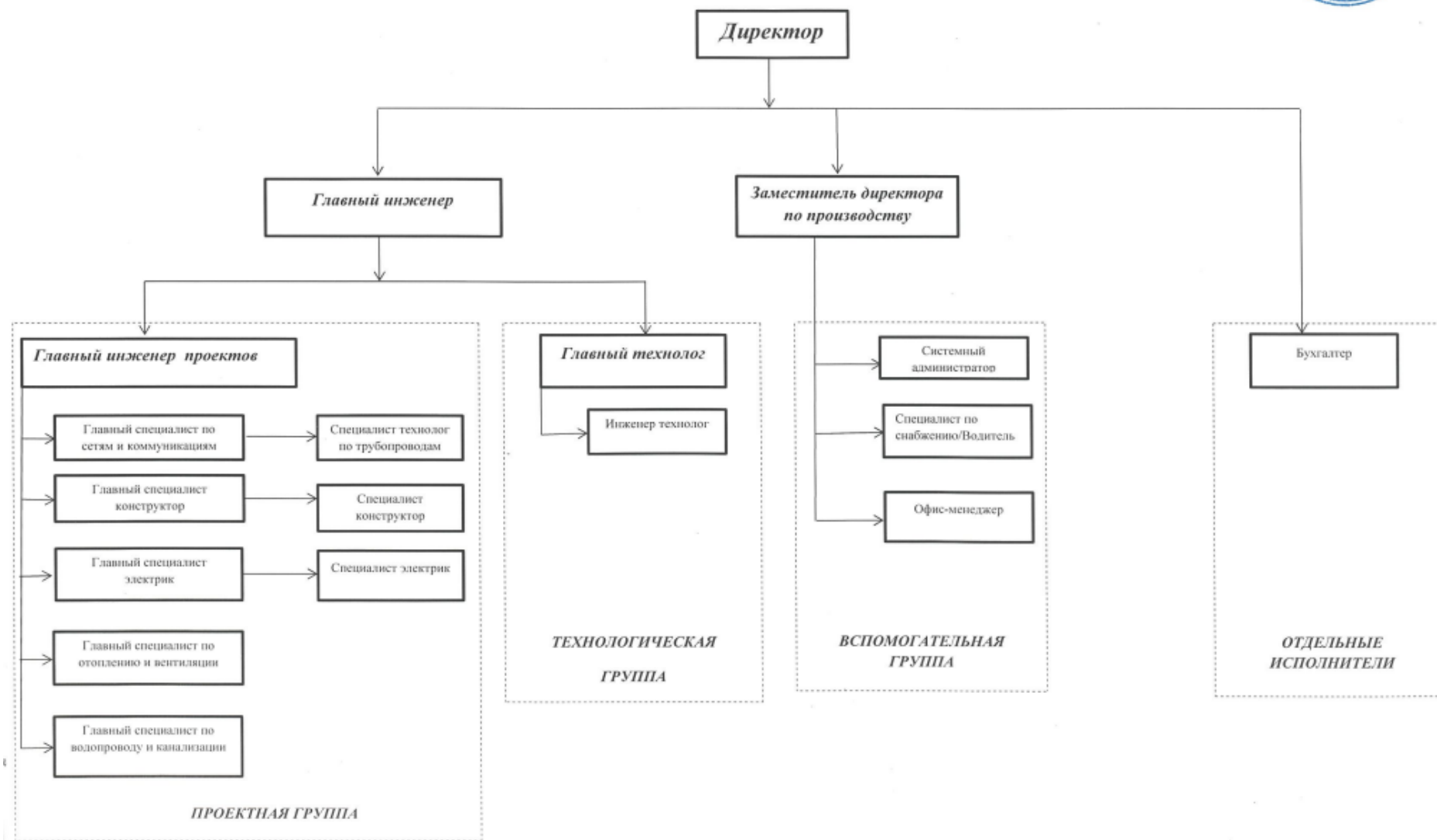


Рисунок 1.15 – Организационная структура компании

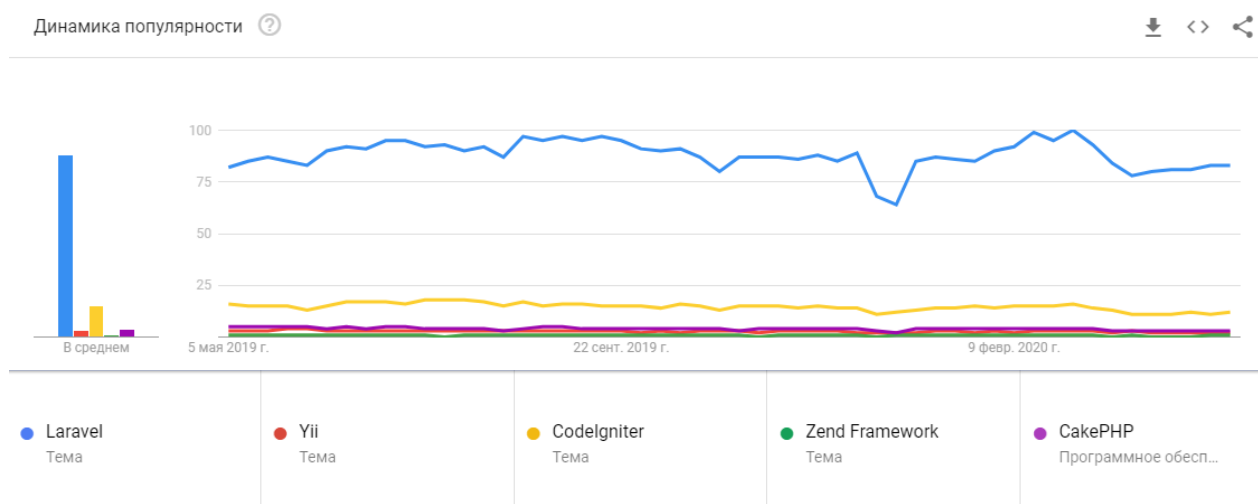


Рисунок 1.16 – Статистика популярности по Google Trends

Вывод. Таким образом в ходе изучения данных фреймворков и специфики их работы, можно сделать выводы о том, что такая связка языков программирования и фремворков обеспечит простоту и удобство для самого разработчика, а также со стороны клиента сократит затраты на разработку и сопровождение ПО, так как используются популярные решения с открытым исходным кодом.

1.7 Вывод

В ходе написания аналитического части дипломного проекта были изучены современные направления развития проект-менеджмента, а также разработки веб-приложений. На сегодняшний день существует огромное количество методик управления проектированием, при комбинировании которых, возможно создание своей собственной системы управления, которая будет отвечать требованиям и соответствовать специфике работы компании. Со стороны программной реализации такой системы существует также огромное количество решений. Для реализации были выбраны современные решения, позволяющие масштабировать виды процессов, подлежащих автоматизации, а также обеспечивающие постоянное развитие и улучшение системы.

2. Разработка информационного обеспечения системы

2.1 Постановка задачи

Любая постановка задачи складывается из этапов:

- формулировка цели;
- мысленное моделирование, т.е. представление путей решения задачи, формулировка результатов, которых нужно добиться;
- словесное (лингвистическое) описание на бумаге входной (исходные данные) и выходной (результаты решения) информации;
- формализованное (математизированное) описание;
- формулировка критериев эффективности работы системы;
- алгоритмизация решения, т.е. описание алгоритма действий, которые производятся с входной информацией, чтобы получить выходную[10].

Согласно алгоритму постановки задачи, цель информационной системы: информационная система для организации «ЭлитСтройПроект-КС» разрабатывается с целью автоматизации и упрощения процесса управления разработкой проекта и работой проектной группы.

Назначение информационной системы:

- обеспечения сбора и первичной обработки исходной информации, необходимой для подготовки отчетности по показателям деятельности;
- создания единой системы отчетности по показателям деятельности;
- повышения скорости и качества работы сотрудников.

Информационная система предназначена для упрощения процесса принятия решений по управлению проектно-строительной организации. Основное назначение – отображение состояния проектных работ, а именно информация об объекте, стадиях проектирования и строительства. ИС позволяет узнать какие работники работают над указанным проектом и на какой стадии они находятся, что позволяет оценить скорость и качество работы отдельных сотрудников.

Основной объект автоматизации – строительные проекты, а точнее информация о них, которая бы позволила получить полное представление о состоянии работ по этому проекту. Таким образом, результат автоматизации должен представлять собой краткую, но емкую информацию о проекте, которая позволит судить о том, на каком этапе находится проект, кто из сотрудников работает над проектом и насколько качественно.

Функциональные возможности разрабатываемой системы:

- возможность создания карточек проектов и постановки задач по проектам;
- назначение задачи определенному сотруднику;
- определения сроков и приоритета задачи;
- получение информации по проектам, позволяющей судить о прогрессе по этому проекту.

При помощи данной системы, руководство организации сможет оценивать прогресс по разработке определенного проекта, назначать на этот

проект кураторов, которые определяют необходимые для выполнения задачи и посредством этой же системы смогут назначать на эти задачи сотрудников организации, в соответствии с их должностями, опытом работы и навыками.

2.2 Проектирование функциональной структуры ИС

В ходе анализа предметной области были выделены сущности, которые можно отобразить как подсистемы ИС. Тогда функциональная структура (см. рис. 2.1) информационной системы будет состоять из:

- подсистема «Задачи» – позволяет просматривать задачи, назначенные текущему пользователю, а также добавление новых задач и назначения пользователей на задачу. Помимо этого, в ходе выполнения задачи возможно изменение ее стадии, приоритета, а также сроков исполнения;

- подсистема «Проекты» – отображает проекты, в которых участвует текущий пользователь, позволяет создать новый проект. При создании нового проекта необходимо заполнить карточку с основной информацией по проекту, после создания можно добавлять задачи по проекту.



Рисунок 2.1 – Функциональная структура ИС

2.3 Проектирование бизнес-модели и оптимизации бизнес-процессов

2.3.1 Функциональная модель

Управление проектами в данной организации осуществляется посредством электронной почты, в печатной, либо в устной форме. То есть задачи, необходимые для выполнения в ходе разработки проекта, сообщаются лично, посредством электронных писем, либо напечатанные на бумаге. Все файлы и документы по проектам хранятся на сетевом диске. Функциональная модель представлена на рисунке 2.2.

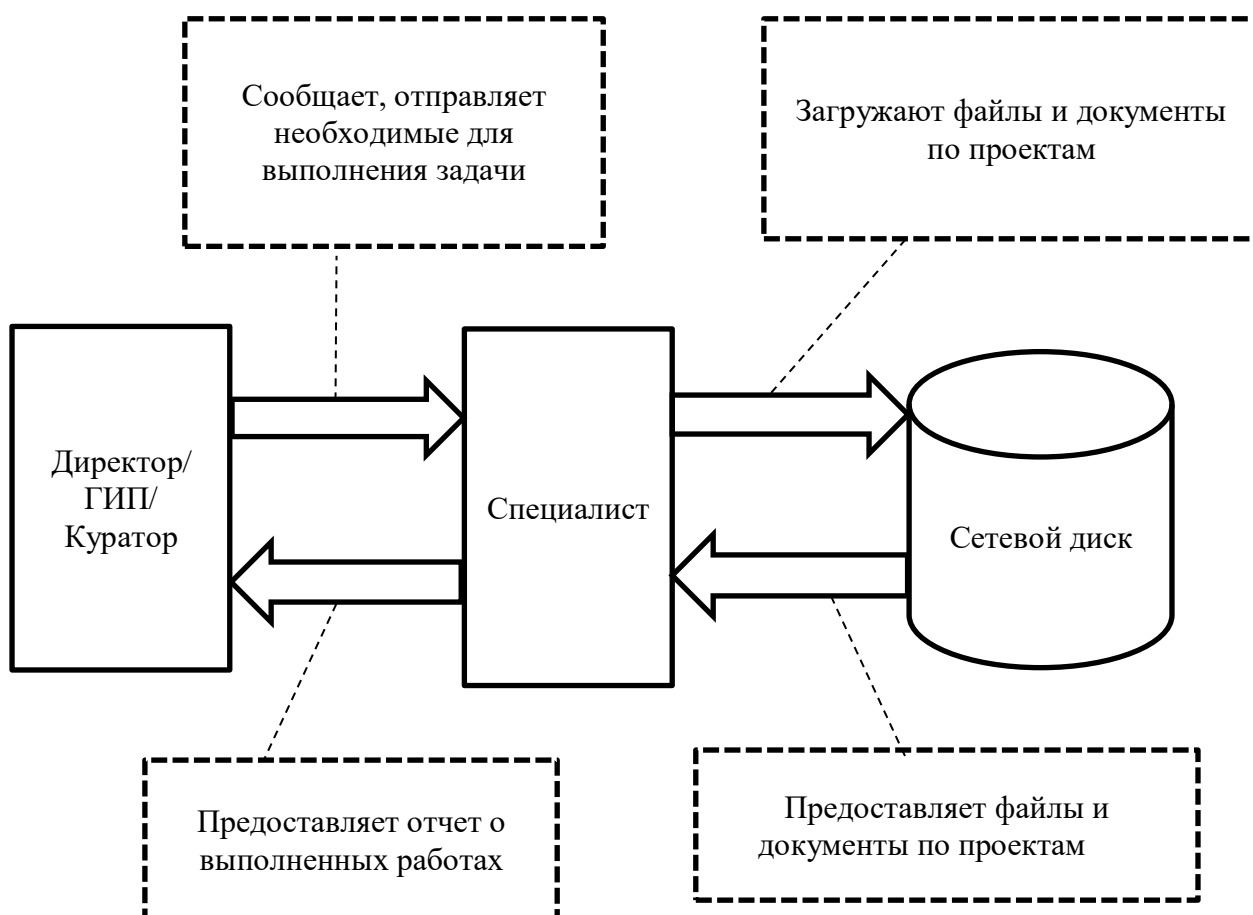


Рисунок 2.2 – Функциональная модель

2.3.2 Информационная модель

Информационная модель отображена на рисунке 2.3. Она отображает то, как и посредством чего передается информация в рассматриваемой предметной области. Первый этап – это получение руководством заказа на проектирование объекта, руководство формирует задание. После чего задание направляется исполнителю в виде электронного письма или передается в устной форме. Всю дополнительную информацию (топосъемка, геологический из), на которой основывается проектирование, проектная группа получает также посредством электронной почты, в распечатанном виде или в устном. Далее они проектируют чертежи, на основе которых

сметный отдел рассчитывает сметы. В дальнейшем проект отправляется на экспертизу, после прохождения которой обрабатывает замечания. Результатом успешной разработки проекта может служить проект организации строительства, по которому уже и строят объект.

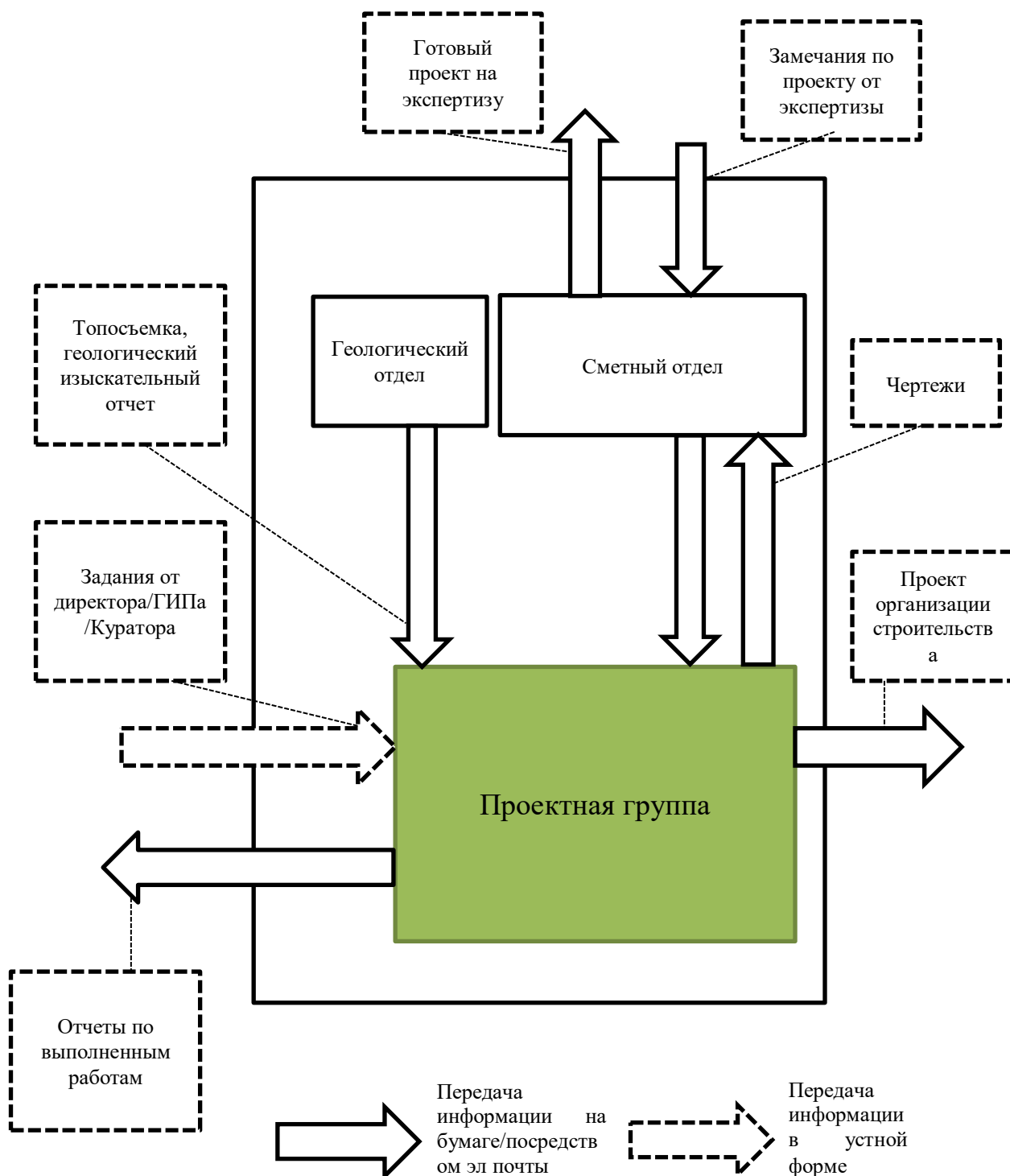


Рисунок 2.3 – Информационная модель

2.3.3 Определение миссии, критических факторов успеха

Миссия организации – успешная разработка проекта, отвечающего всем требованиям заказчика.

Критические факторы успеха:

- качественная разработка проектов, отвечающих требованиям заказчика и при этом, соответствующих государственным нормативам РК;
- скорость и минимальные информационные потери при обмене информацией между директором/ГИПом/куратором и специалистами проектной группы;
- эффективная система отчетности перед Директором и заказчиками.

Структурные проблемы:

- передача информации между управляющими и управляемыми проходит по электронной почте, в устной форме, что замедляет, либо повышает риск непонимания выполняемой задачи;
- отсутствие возможности оценки состояния проекта;
- отсутствие организации файлов по проектам.

Цели информационной системы:

Повысить эффективность и удобство разработки проектов и оценки их состояния.

Задачи информационной системы

- создать базу данных, содержащую сведения, необходимые для управления проектами;
- создать эргономичный пользовательский интерфейс, отображающий проекты, их состояние и выполняемые по ним задачи, назначенные определенным специалистам.

2.3.4 Выделение бизнес-процессов.

В работе выделяются следующие бизнес-процессы: «Получение проекта» (рис. 2.4), «Назначение задач специалистам» (рис. 2.5), «Выполнение задачи» (рис. 2.6), «Контроль разработки проекта» (рис. 2.7).

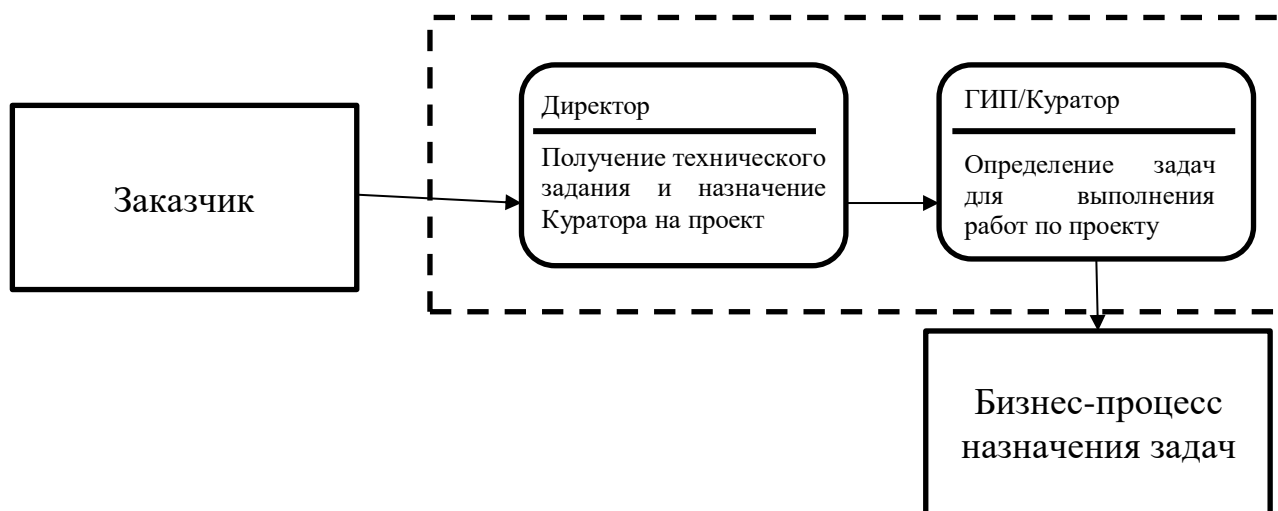


Рисунок 2.4 – Бизнес-процесс «Получение проекта»

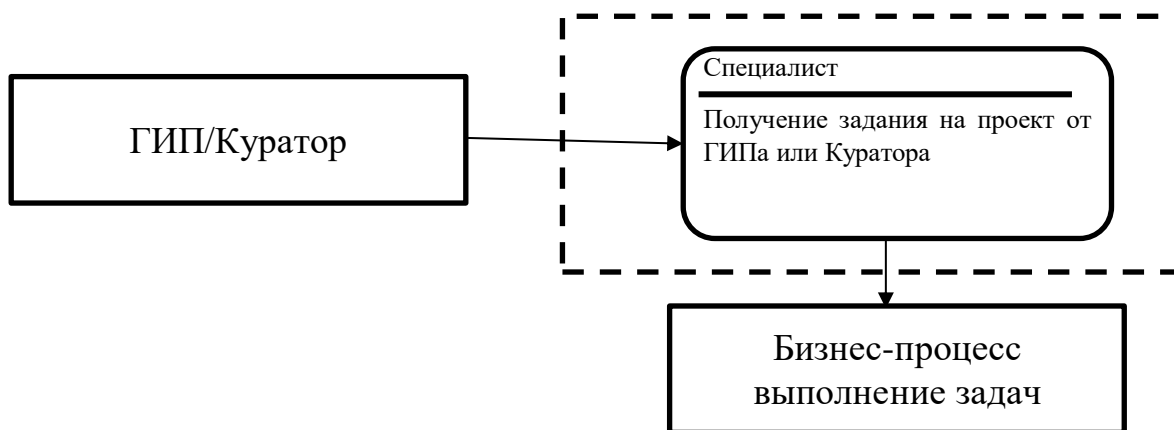


Рисунок 2.5 – Бизнес-процесс «Назначение задач»

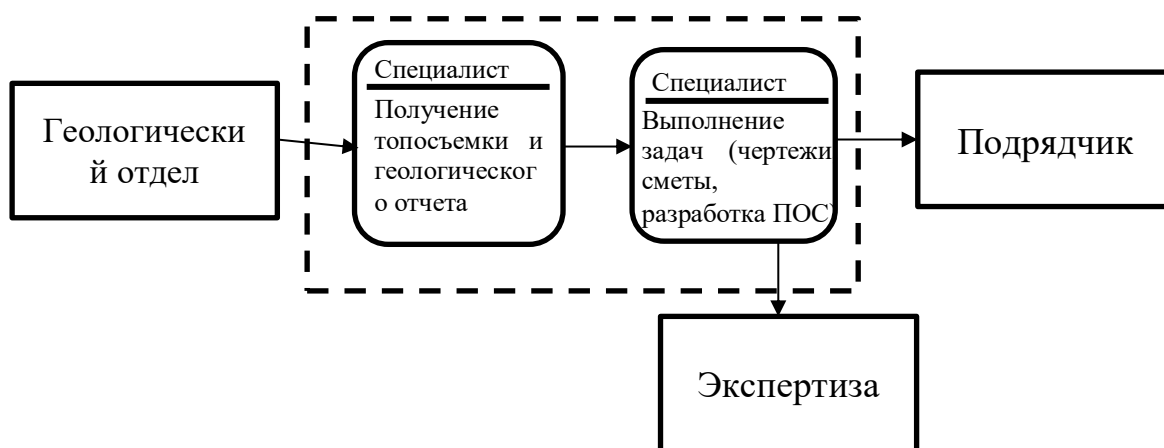


Рисунок 2.6 – Бизнес-процесс «Выполнение задачи»

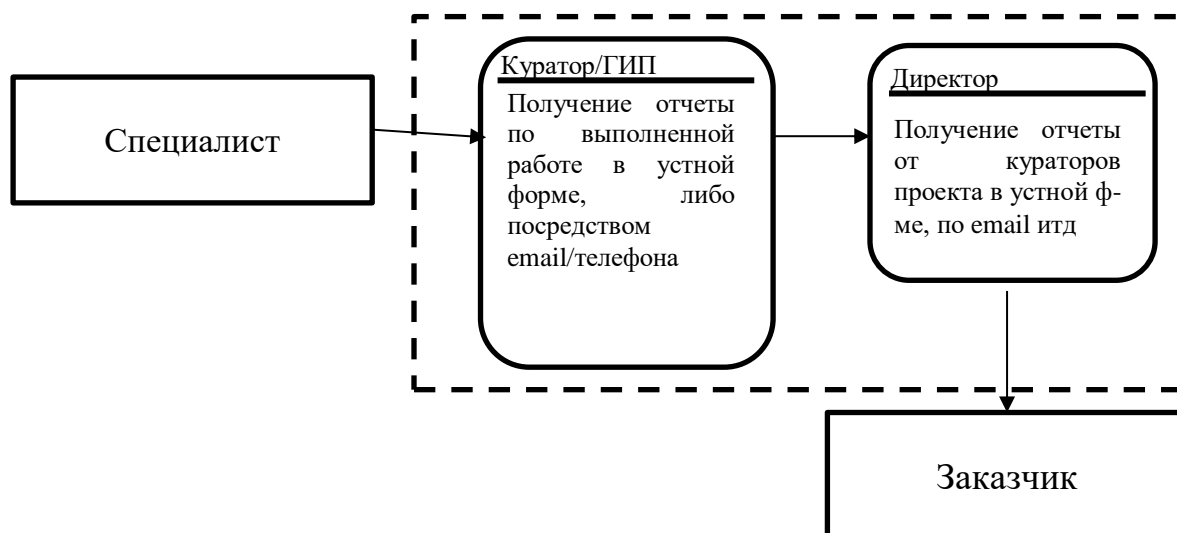


Рисунок 2.7 – Бизнес-процесс «Контроль разработки проекта»

2.3.5 Анализ и оптимизация бизнес-процессов.

Оценка критических факторов успеха управления проектами представлена в таблице 2.1.

Таблица 2.1 – Оценка критических факторов успеха

Бизнес-процессы	Качественная разработка проекта	Надежность обмена данными между управляющими и управляемыми	Отчетность
Получение проекта	0	0	0
Назначение задач	4	3	0
Выполнение задач	5	3	5
Контроль разработки проекта	4	2	5

Для оптимизации системы управления проектами создается база данных для обмена данными между управляющими и управляемыми. Оптимизированные с ее помощью бизнес-процессы показаны ниже.

Оптимизация «Получения проекта» представлена на рисунке 2.8.

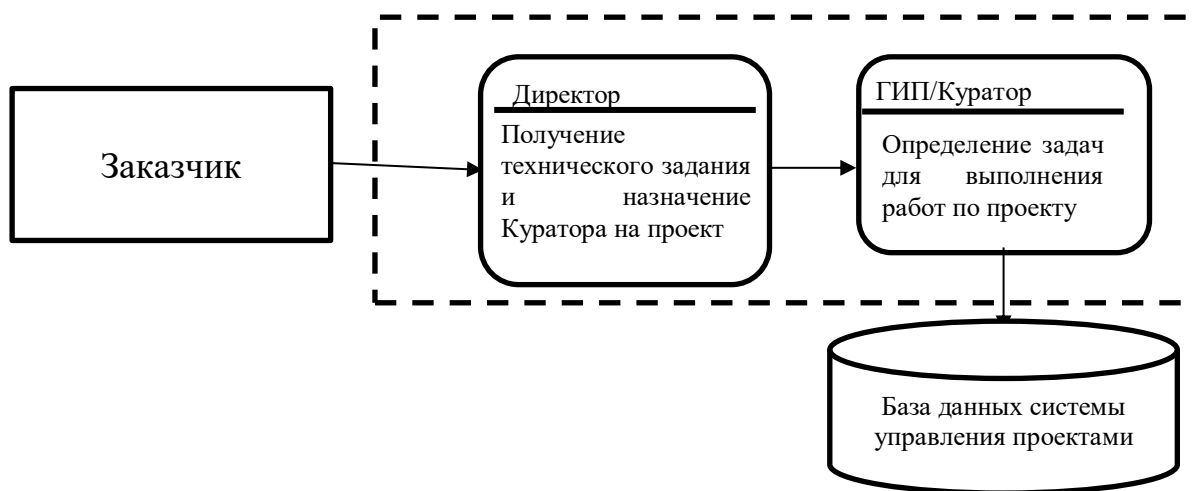


Рисунок 2.8 – Оптимизированный бизнес-процесс «Получения проекта»

Оптимизация «Назначения задач по проекту» представлена на следующем рисунке 2.9.

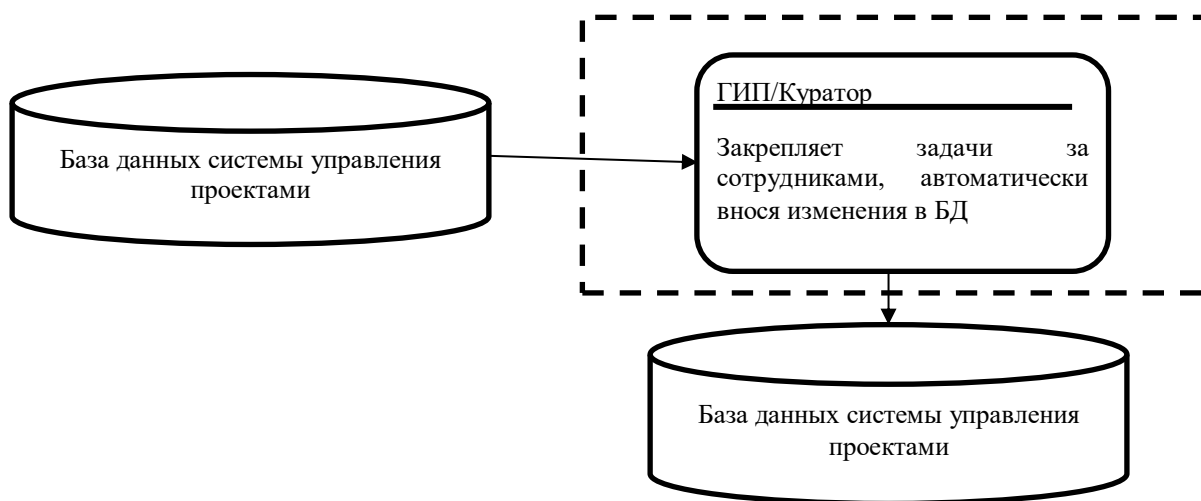


Рисунок 2.9 – Оптимизированный бизнес-процесс «Назначения задач по проекту»

Оптимизация бизнес-процесса «Выполнения задач» отображена на рисунке 2.10, а «Контроля разработки проекта» на рисунке 2.11

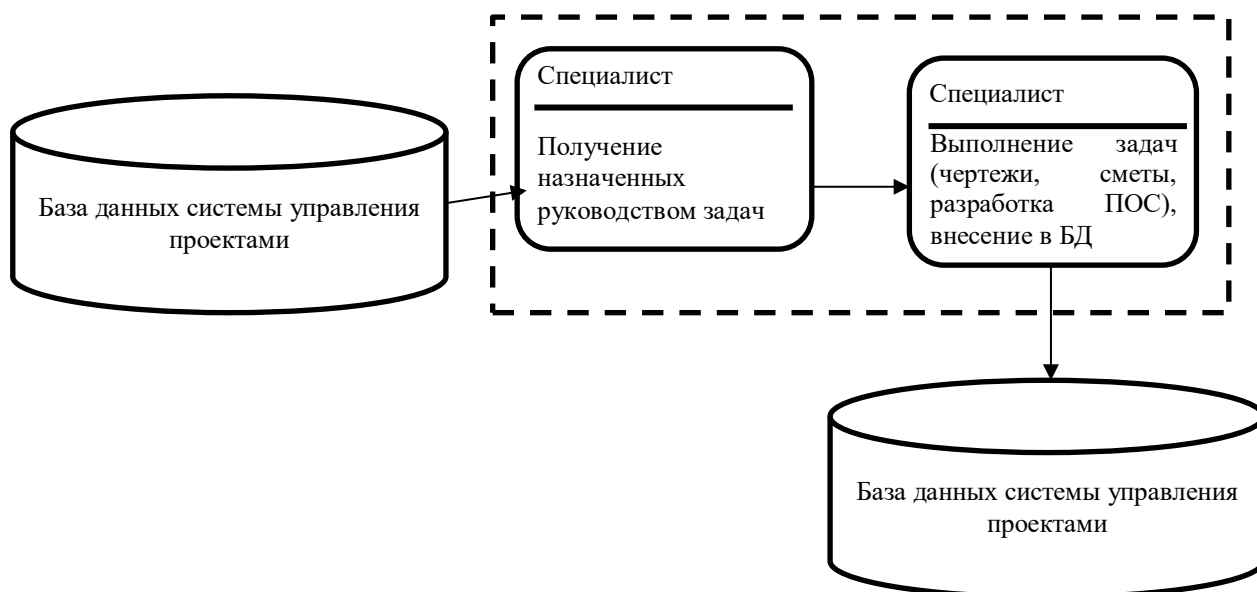


Рисунок 2.10 – Оптимизированный бизнес-процесс «Выполнения задач»

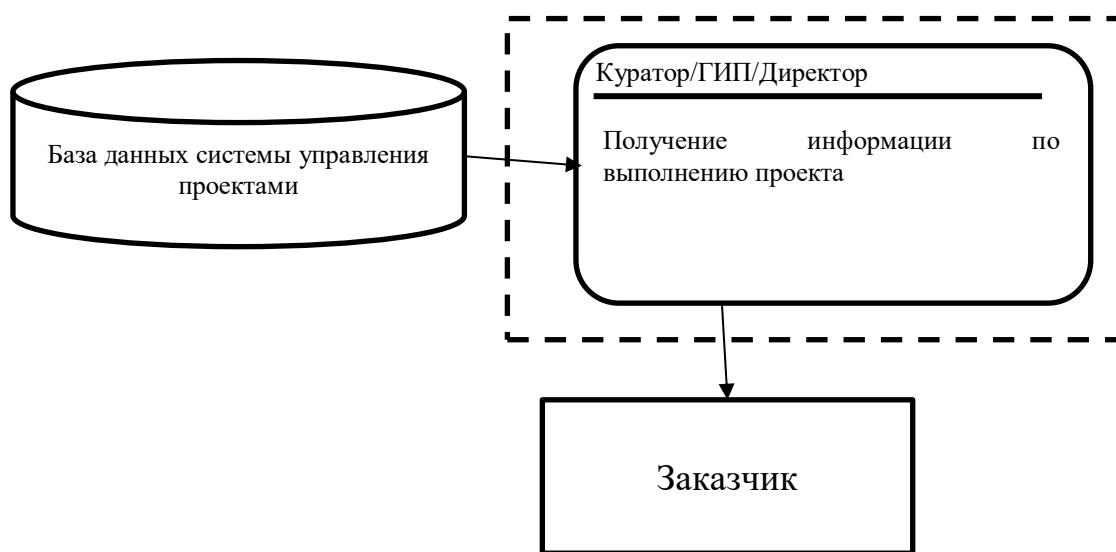


Рисунок 2.11 – Оптимизированный бизнес-процесс «Контроля разработки проекта»

2.4 Проектирование информационного обеспечения

2.4.1 Обоснование выбора СУБД

а) MySQL – на сегодня это одна из самых распространенных СУБД. Обычно используется при разработке веб-сайтов и других веб-приложений. Система бесплатна и открыта, что является одним из значительных факторов ее популярности. Активное сообщество пользователей постоянно совершенствует его, разрабатывает собственные плагины, добавляет новые функциональные возможности.

Как правило, большинство современных СУБД для своей работы используют язык запросов SQL. Но MySQL не использует его в чистом виде, то есть некоторые языковые функции отсутствуют.

Преимущества этой системы включают в себя:

- СУБД проста в установке. Как правило, он поставляется в комплекте с веб-сервером, таким как LAMP (Linux, Apache, MySQL, PHP). Программным продуктом можно управлять через консоль или графический интерфейс;
- СУБД практически полностью поддерживает SQL;
- MySQL может довольно просто работать с большими объемами данных и при необходимости легко масштабироваться;
- некоторые упрощения некоторых стандартов языка SQL могут ускорить работу СУБД.

К недостаткам использования MySQL относятся:

- не полное соответствие стандартам SQL. Если до использования этой СУБД специалист работал с системой, использующей стандартный SQL, то при переходе на MySQL могут возникнуть некоторые трудности, связанные с изучением специфики используемого языка;
- проблемы с надежностью. Из-за некоторых методов обработки информации MySQL в некоторых ситуациях уступает другим СУБД. Например: во время обработки кластера MySQL может организовать атаку DDos на свою собственную базу данных.

б) PostgreSQL из всех бесплатных СУБД – это самое профессиональное решение. PostgreSQL – это бесплатная и открытая система. Но в отличие от MySQL, он пытается полностью соответствовать стандартам SQL. Качественно эта СУБД имеет поддержку ACID. Этот набор требований позволяет поддерживать надежные транзакции. Производительность и соответствие СУБД этому набору правил позволяет добиться параллельной работы системы не путем блокировки выполняемых операций, а путем реализации многомерного управления параллелизмом.

Преимущества этой системы включают в себя:

- Postgre – это бесплатная система с открытым исходным кодом, которая соответствует стандартам SQL;
- достаточно большое количество дополнений, позволяющих расширить функционал системы;
- в отличие от других СУБД, он обеспечивает более высокий уровень целостности данных и транзакций.

К недостаткам можно отнести:

- при выполнении довольно простых операций СУБД использует значительно больше ресурсов, чем ее конкуренты;
- несмотря на достаточно большую пользовательскую базу, довольно сложно найти хостинг, который будет поддерживать Postgre.

в) Microsoft SQL Server – разработана Microsoft. Как и MySQL и Postgre, он предназначена для управления реляционными базами данных. Сервер Microsoft SQL использует язык Transact-SQL. Этот язык соответствует

стандартам языка SQL. Эта СУБД может использоваться как с базами данных малых предприятий, так и с базами данных крупных компаний.

Преимущества СУБД включают в себя:

- шифрование всей хранимой информации в базе данных. Шифрование и дешифрование информации происходит непосредственно перед чтением или записью информации;

- интуитивно понятный интерфейс;

- высокая производительность СУБД;

- легкая масштабируемость системы.

К недостаткам можно отнести:

- ограниченный выбор платформ. СУБД будет работать только под управлением операционных систем семейства Windows;

- ограниченное количество инструментов для разработки клиентских приложений. По умолчанию количество платформ для разработки клиентской части ограничено. Другие СУБД поддерживают больше платформ;

- Microsoft SQL Server не является свободным программным обеспечением. SQL-сервер имеет несколько редакций, которые отличаются предоставленной функциональностью и стоимостью.

На рисунке 2.12 Изображена статистика поисковых запросов по данным СУБД за последний год по всему миру.

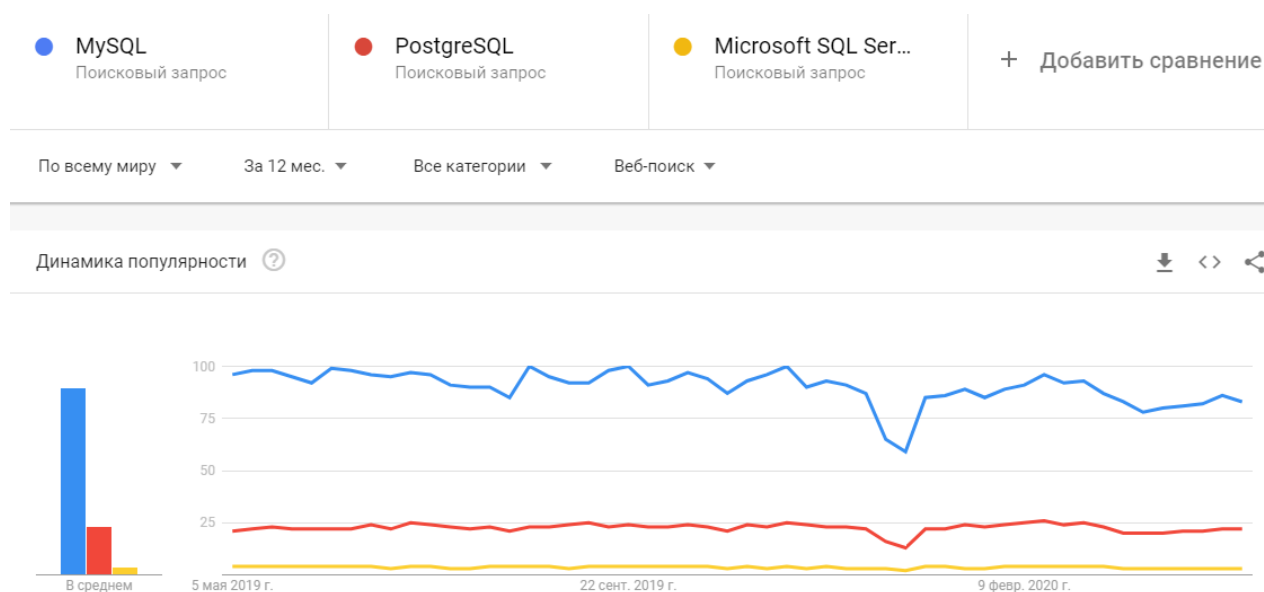


Рисунок 2.12 – Поисковые запросы по СУБД MySQL, PostgreSQL, Microsoft SQL Server

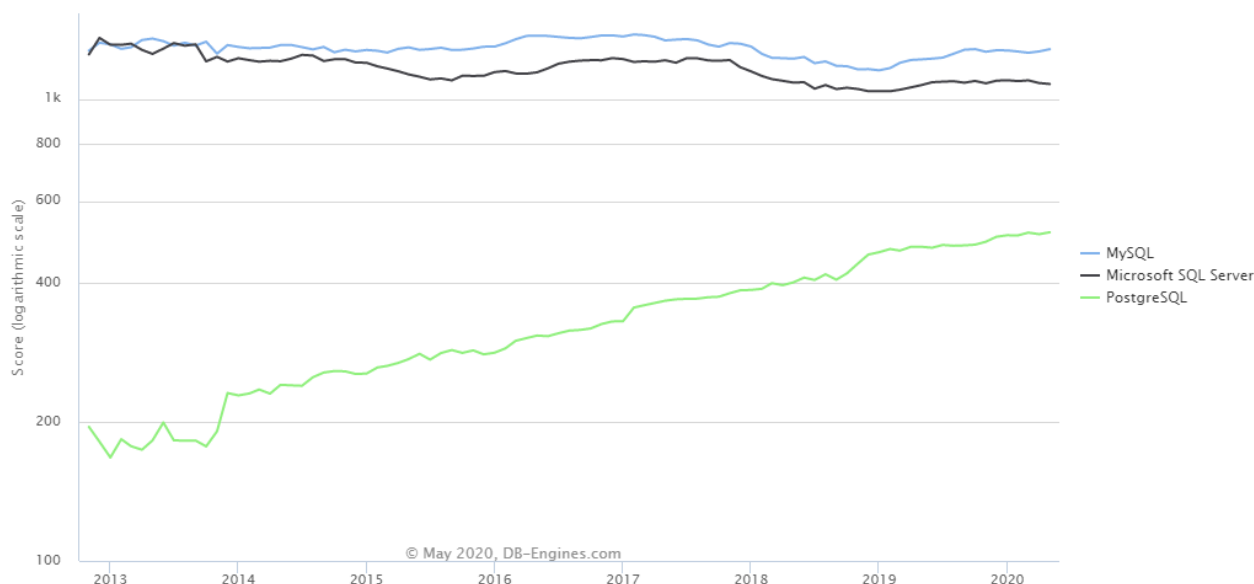


Рисунок 2.13 – Рейтинг популярности СУБД по DB-Engines

Таблица 2.2 – Сравнение системных характеристик[9]

СУБД	Microsoft SQL Server	MySQL	PostgreSQL
Описание	Реляционная СУБД Microsoft	Широко используемая реляционная СУБД с открытым кодом	Широко используемая реляционная СУБД с открытым кодом
Модель базы данных	Реляционная	Реляционная	Реляционная
Лицензия	Платная	Бесплатная	Бесплатная
Серверные ОС	Linux, Windows	FreeBSD, Linux, OS X, Solaris, Windows	FreeBSD, HP-UX, Linux, NetBSD, OpenBSD, OS X, Solaris, Unix, Windows
Схема данных	есть	есть	есть
Типирование	есть	есть	есть
Поддержка XML	есть	есть	есть
Вторичные индексы	есть	есть	есть
API и другие методы доступа	ADO.NET, JDBC, ODBC, OLE DB, TDS	ADO.NET, JDBC, ODBC, Собственный нативный API	ADO.NET, JDBC, Нативная библиотека C, ODBC, потоковый API

Продолжение таблицы 2.2

Поддерживаемые языки программирования	C#, C++, Delphi, Go, Java, JavaScript (Node.js), PHP, Python, R, Ruby, Visual Basic	Ada, C, C#, C++, D, Delphi, Eiffel, Erlang, Haskell, Java, JavaScript (Node.js), Objective-C, OCaml, Perl, PHP, Python, Ruby, Scheme, Tcl	.Net, C, C++, Delphi, Java info, JavaScript (Node.js), Perl, PHP, Python, Tcl
Серверные скрипты	Transact SQL, .NET, R, Python, Java	собственный синтаксис	пользовательские функции
Триггеры	есть	есть	есть
Методы разбиения	таблицы могут быть распределены по нескольким файлам (горизонтальное разбиение)	горизонтальное разбиение, разделение с помощью MySQL Cluster или MySQL Fabric	разбиение по диапазону, списку и (начиная с PostgreSQL 11) по хешу
Внешние ключи	есть	есть	есть

Вывод. Исходя из вышеперечисленных преимуществ и недостатков популярных СУБД, выбор пал на СУБД MySQL. Также на выбор данной СУБД повлияло то, что эта СУБД уже была изучена, поэтому будет легче работать с ней. К тому же, данная СУБД используется в организации, для которой пишется данная система управления проектами. Помимо этого, СУБД MySQL является бесплатной, а также, в основном, используется при разработке веб сайтов и других веб приложений. В качестве интерфейса для администрирования СУБД выбрано веб-приложение phpMyAdmin, которое позволяет запускать SQL-команды, просматривать содержимое баз данных их таблиц, а также осуществлять администрирование.

2.4.2 Анализ предметной области

При взаимодействии с разрабатываемой ИС, сотрудник организации – один из пользователей, может просматривать и добавлять задачи по проекту, комментарии к этим задачам, а также файлы по проектам, в случае необходимости также возможно разделение существующих задач на подзадачи. Для визуализации таких взаимодействий используются UML-диаграммы прецедентов (вариантов использования). Такие диаграммы отражают отношения между актерами и прецедентами, позволяют определить

функциональные требования к системе и описать отношения между пользователями системы и самой системой.

Диаграмма вариантов использования для подсистемы «Задачи» пользователем изображена на рисунке 2.14



Рисунок 2.14 – Диаграмма прецедентов для актора «Пользователя» в подсистеме «Задачи»

Диаграмма прецедентов для подсистемы «Проекты» представлена на рисунке 2.15

Диаграмма прецедентов для всей системы отображает все взаимодействия пользователя и системы (рис 2.16).

Выполняемые пользователем действия, для просмотра информации по проекту изображены на рисунке 2.17. Для прецедента «Просмотр информации по проекту» выполняются следующие действия:

- аутентификация (ввод логина и пароля пользователя);
- переход на вкладку «Проекты»;
- выбор интересующего проекта;
- просмотр информации.

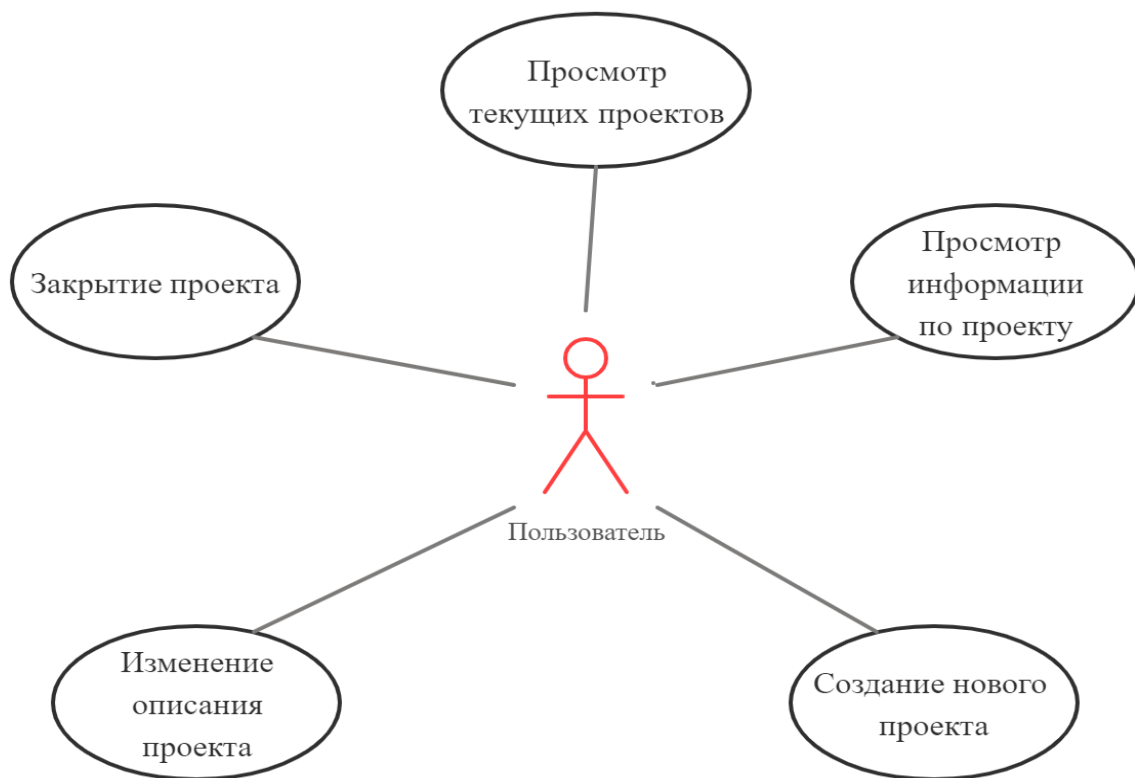


Рисунок 2.15 – Диаграмма прецедентов для актора «Пользователя» в подсистеме «Проекты»

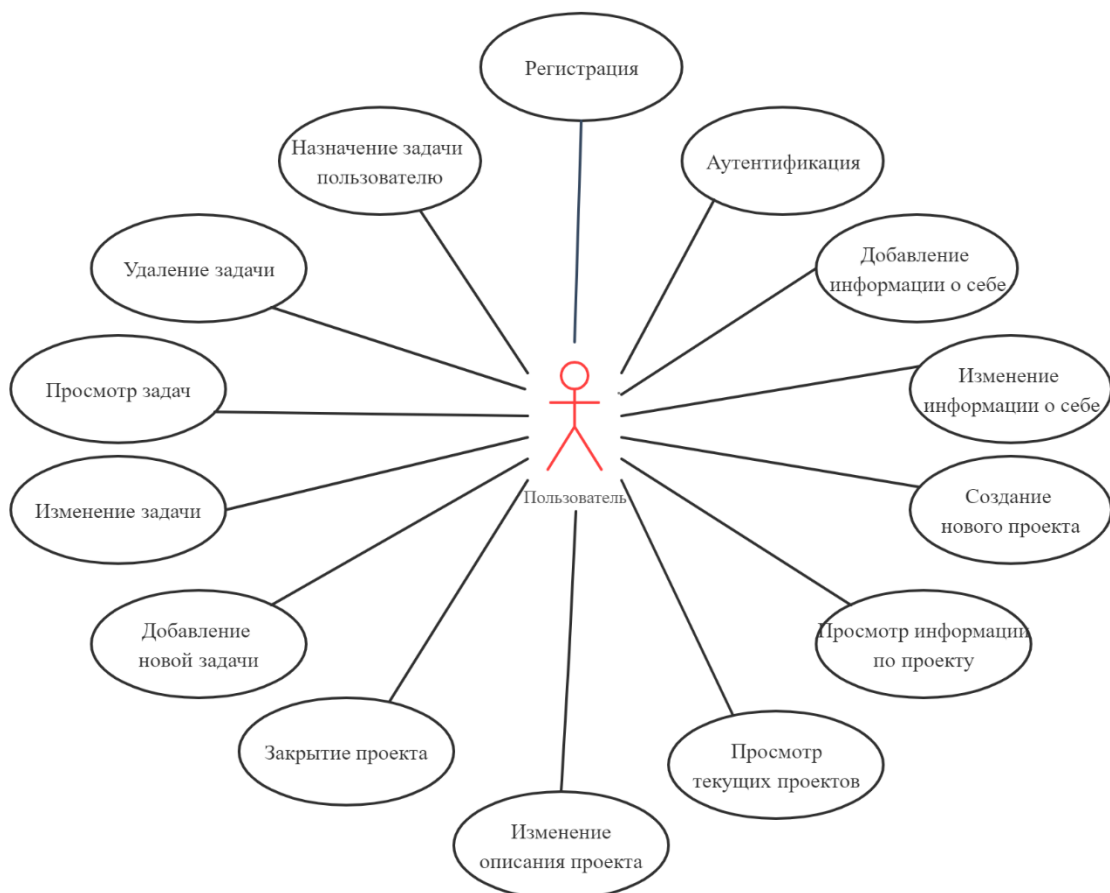


Рисунок 2.16 – Диаграмма прецедентов для актора «Пользователя»

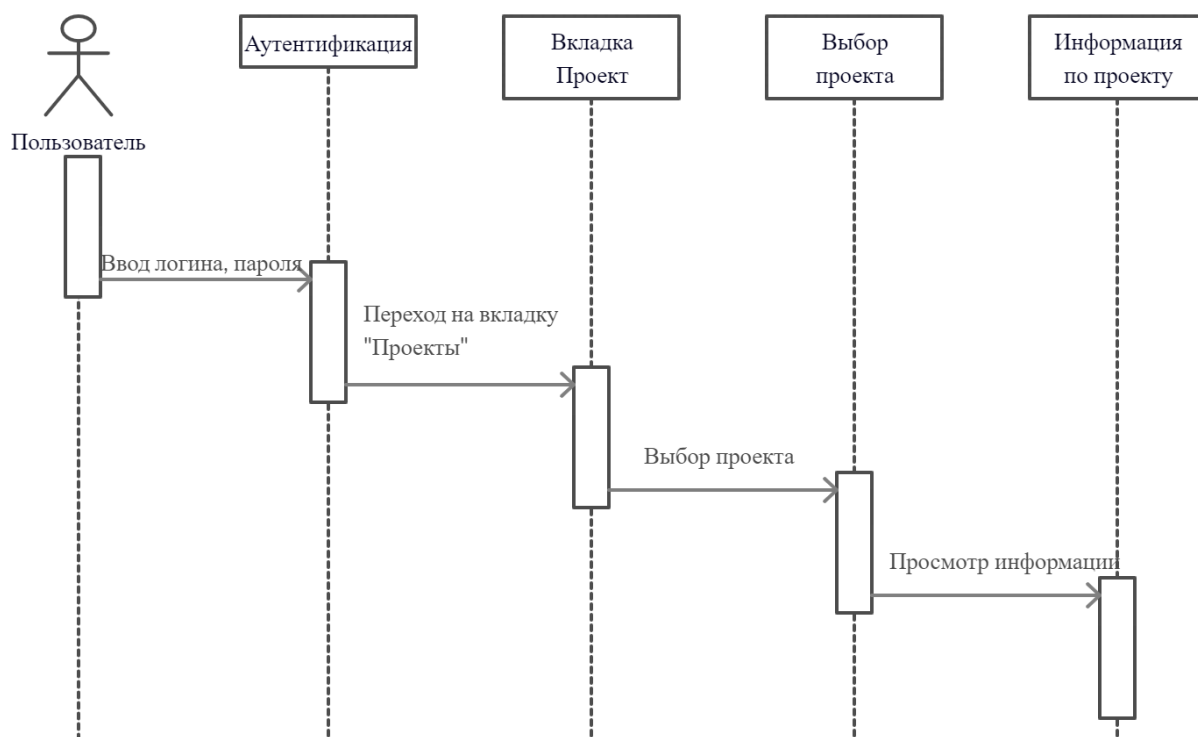


Рисунок 2.17 – Диаграмма последовательности варианта использования «Просмотр информации по проекту»

Учитывая бизнес-модель ИС управления проектами, можно выделить данные подлежащие хранению. К ним относятся личные данные пользователей ИС (ФИО, должность и т.д.), информация по проектам (название, сроки начала и окончания проектирования, описание, файлы и документация по проекту, выполняемые задачи), информация по задачам (кому назначена, подзадачи, описание).

Для проектирования базы данных используется метод “сущность-связь”. Первым шагом в процессе проектирования баз данных является выделение сущностей, их атрибутов и связей между сущностями.

В ходе анализа предметной области были выделены такие сущности как:

- сотрудник;
- должность;
- проект;
- задача.

Атрибуты сущности «Сотрудник»:

- ФИО;
- должность;

Атрибуты сущности «Проект»:

- название;
- срок начала проектирования;
- срок окончания проектирования;
- описание проекта;
- файлы по проекту;

- документы по проекту;
- выполняемые задачи;
- заказчик.

Атрибуты сущности «Задачи»:

- название задачи;
- описание;
- должности, рекомендуемые для выполнения этой задачи;

Атрибуты сущности «Должность»:

- название должности;
- описание.

2.4.3 Описание структур таблиц

База данных информационной системы управления проектами состоит из четырех таблиц:

- сотрудник;
- должность;
- проект;
- задача.

Таблица «Сотрудник» содержит информацию о сотрудниках организации, а именно его ID, ФИО, должность, ее структура показана в таблице 2.3.

Таблица 2.3 – Структура таблицы «Сотрудник»

	Наименование поля	Тип поля	Размер	Назначение
1	ID_сотрудника	INTEGER	4 BYTE	Является первичным ключом, а также хранит идентификатор сотрудника
2	Логин	VARCHAR2	50 BYTE	Хранит логин для входа в систему
3	Фамилия	VARCHAR2	70 BYTE	Хранит сведения о фамилии
4	Имя	VARCHAR2	70 BYTE	Хранит сведения об имени
5	Отчество	VARCHAR2	70 BYTE	Хранит сведения об отчестве
6	Код должности	INTEGER	4 BYTE	Хранит сведения о должности сотрудника

Таблица «Проект» содержит информацию о конкретном проекте, а именно его код, название, описание, сроки проектирования, заказчика, задачи необходимые для выполнения в ходе проектирования, ее структура показана в таблице 2.4.

Таблица 2.4 – Структура таблицы «Проект»

	Наименование поля	Тип поля	Размер	Назначение
1	Код проекта	INTEGER	4 BYTE	Является первичным ключом и хранит

				идентификатор проекта
Продолжение таблицы 2.4				
2	Название проекта	VARCHAR2	70 BYTE	Хранит сведения о названии проекта
3	Описание	VARCHAR2	500 BYTE	Описание проекта
4	Начало	DATE		Хранит сведения о начале работы над проектом
5	Окончание	DATE		Хранит сведения об окончании работы над проектом
6	Заказчик	VARCHAR2	100 BYTE	Хранит сведения о заказчике
7	Код задачи	INTEGER	4 BYTE	Хранит сведения о задаче, выполняемой для проектирования

Таблица «Задача» содержит информацию о конкретной задаче, а именно её код, название, описание, рекомендуемые для этой задачи должности и проект, к которому относится эта задача, ее структура показана в таблице 2.5.

Таблица 2.5 – Структура таблицы «Задача»

	Наименование поля	Тип поля	Размер	Назначение
1	Код задачи	INTEGER	4 BYTE	Является первичным ключом и хранит идентификатор задачи
2	Название задачи	VARCHAR2	70 BYTE	Хранит сведения о названии задачи
3	Описание	VARCHAR2	400 BYTE	Описание задачи
4	Код проекта	INTEGER	4 BYTE	Хранит сведения о проекте, к которому относится задача

Таблица 2.6 – Структура таблицы «Должность»

	Наименование поля	Тип поля	Размер	Назначение
1	Код должности	INTEGER	4 BYTE	Является первичным ключом и хранит идентификатор должности
2	Название	VARCHAR2	70 BYTE	Хранит сведения о названии должности
3	Описание	VARCHAR2	4 BYTE	Описание должности

2.4.4 ER-модель данных

Нотация IDEF1X основана на подходе Чена. Сущность в такой нотации независима от идентификаторов только в том случае, если каждый экземпляр такой сущности однозначно идентифицирован без определения отношений с другими сущностями. А зависимой считается сущность, соответственно, зависимая от связей.

Зависимые сущности представлены двумя группами: родители и потомки. Родительская сущность ассоциирована с произвольным количеством экземпляров сущности-потомка. А каждый экземпляр потомка ассоциирован только с одним экземпляром сущности-родителя. Связь между родителем и потомком изображается линией, ведущей от родителя к потомку, и заканчивающейся точкой.

Связь бывает идентифицирующей и неидентифицирующей. Идентифицирующая связь означает то, что потомок зависим от идентификатора. Изображается сплошной линией.

Сущность-потомок в неидентифицирующей связи не зависим от идентификатора в том случае, если родительская сущность также не является потомком идентифицирующей связи. Изображается пунктирной линией.

Атрибуты сущности отображаются в блоке сущности, наверху блока располагаются атрибуты – первичные ключи и отделены от остальных атрибутов линией. Помимо первичных, сущности имеют также внешние ключи, в скобках указывается FK – foreign key.

Также связь может иметь мощность:

- N – экземпляр родителя может иметь 0, 1 или больше одного потомка;
- P – родитель должен иметь не менее одного потомка;
- Z – родитель должен иметь не более одного потомка;
- M – родитель связан с фиксированным числом потомков.

ER – модель представлена на рисунке 2.18. А физическая структура изображена на рисунке 2.19.

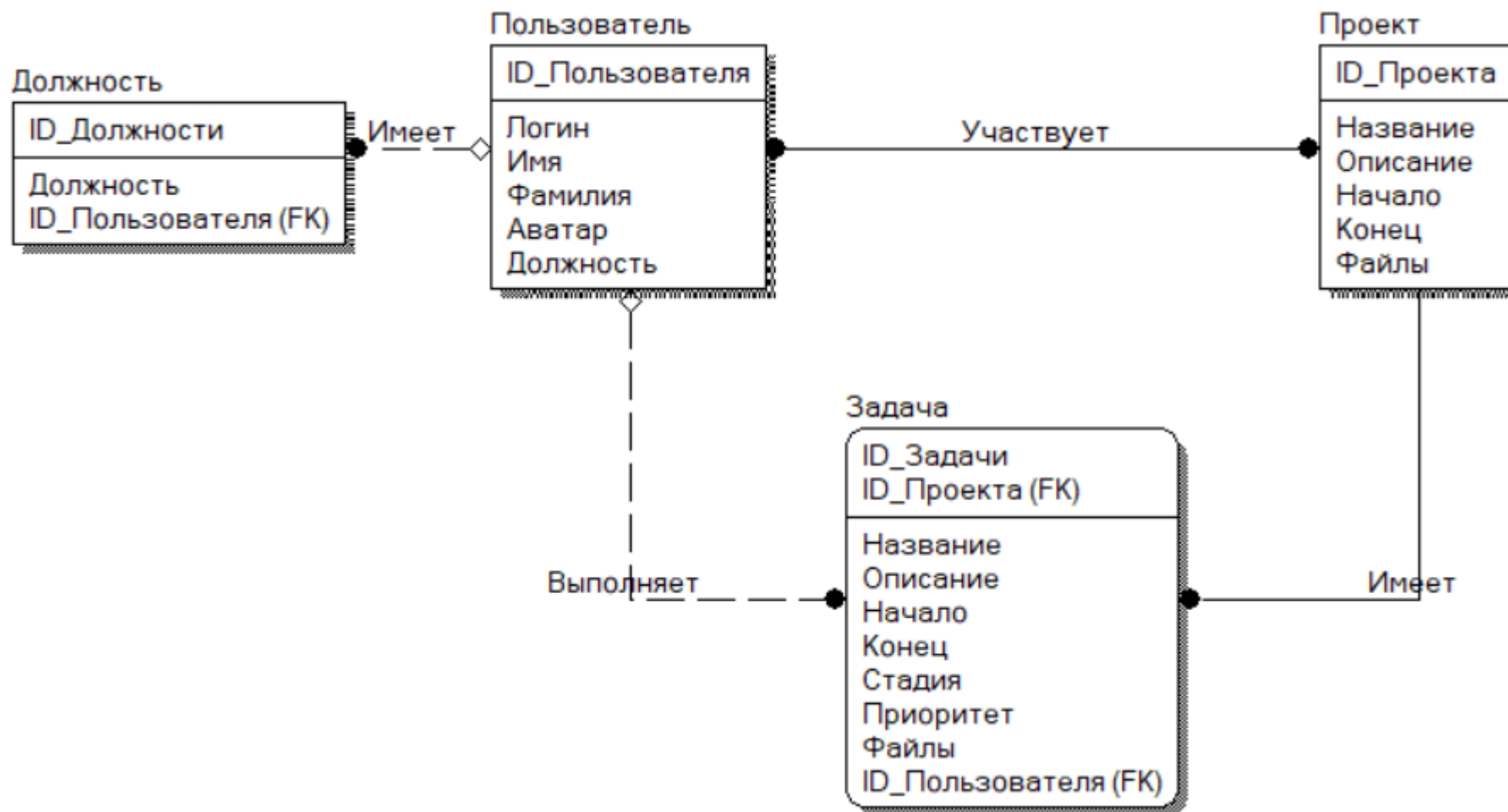


Рисунок 2.18 – ER-модель базы данных системы

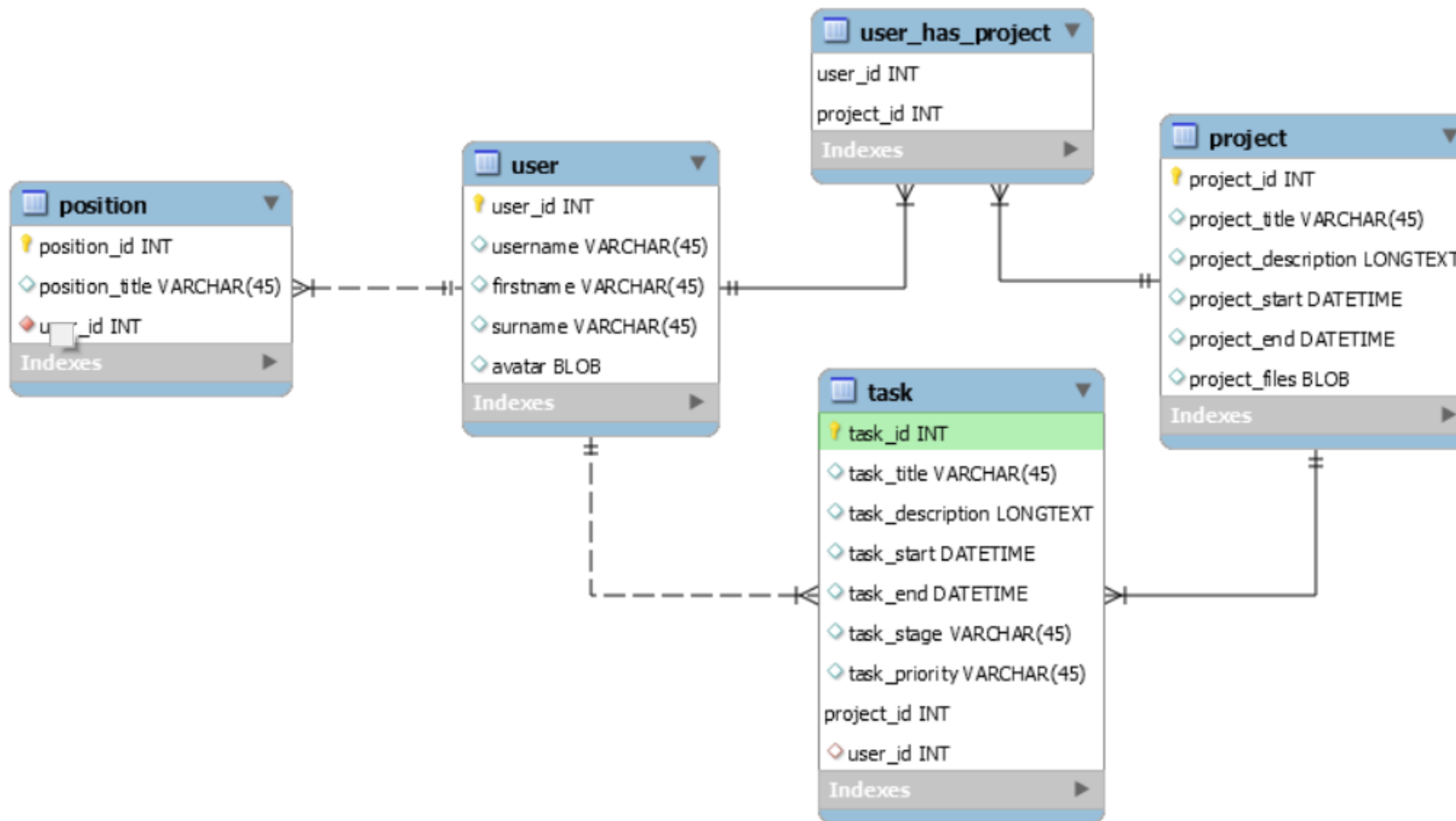


Рисунок 2.19 – Физическая модель базы данных системы

3 Разработка программного обеспечения системы

3.1 Разработка веб-приложения

3.1.1 Настройка базы данных

Для того чтобы создать Laravel-проект необходимо установить PHP версией не менее 7.2.5, а также менеджер зависимостей для PHP – Composer. Для установки Laravel используется команда, которая позволяет установить его глобально:

```
composer global require laravel/installer
```

Для создания проекта используется команда:

```
composer create-project --prefer-dist laravel/laravel [название проекта]
```

Чтобы связать проект с определенной БД в .env-файле необходимо изменить первоначальные значения некоторых переменных.

Для работы с базами данных в Laravel используются модели Eloquent, которые позволяют запрашивать данные из таблиц и вставлять в таблицы новые записи. Для создания модели используется команда:

```
php artisan make:model [название модели]
```

Для определения структуры таблиц базы данных в Laravel используются миграции. Вместо того, чтобы вручную добавлять столбцы в локальные копии БД, можно просто запустить миграции. Для создания миграции используется команда - `php artisan make:migration [название миграции]`. Сама миграция выглядит следующим образом см. листинг 3.1

Листинг 3.1 – Миграция для создания таблицы «Должность»

```
class CreatePositionsTable extends Migration
{
    public function up()//добавляет
    {
        Schema::create('positions', function (Blueprint $table) {
            $table->id();
            $table->string('name');
            $table->timestamps();
            $table->softDeletes();
        });
    }
    public function down()//делает откат
    {
        Schema::dropIfExists('positions');
    }
}
```


Чтобы запустить миграцию используется команда - `php artisan migrate`. Она создает все таблицы БД.

Для получения и хранения данных в базе данных используются модели Eloquent. Каждая Eloquent модель однозначно соответствует одной таблице базы данных.

Eloquent ORM – это система объектно-реляционного отображения в которой каждая таблица имеет свой класс-модель при помощи которого осуществляется работа с этой таблицей – модель может запрашивать данные, а также вставлять новые данные в таблицу. Модели Eloquent можно воспринимать как конструктор запросов. Модель привязывается к таблице, которое соответствует имени самой модели, но в нижнем регистре и во множественном числе, т.е. модель `Position` автоматически привяжется к таблице `positions`. Что касается первичных ключей, то Eloquent ожидает, что каждая таблица имеет первичный ключ с именем `id` и является инкрементным числом. А также Eloquent предполагает, что в таблице будут поля `updated_at` и `created_at` (время изменения и создания).

В листинге 3.2 представлен фрагмент программы, в котором описана модель задания (`Task`), с помощью которой реализуются запросы к базе данных.

Листинг 3.2 – Eloquent модель таблицы «Задания»

```
<?php
namespace App;
use App\DB;
use Illuminate\Database\Eloquent\Model;
class Task extends Model
{
    protected $table = 'tasks';
    protected $primaryKey = 'id';
    public $incrementing = TRUE;
    public $timestamps = FALSE;
    public function getTasks() {
        $tasks = Task::all();
    }
    public function getProjectTasks() {
        $tasks = DB::table('tasks')
            ->join('project_tasks', 'tasks.id', '=', 'project_tasks.task_id')
            ->join('projects', 'project_tasks.project_id', '=', 'project.id')
            ->select('projects.title', 'tasks.title', 'tasks.start', 'tasks.end')
            ->get();
    }
    public function getUserTasks() {
```

```

$tasks = DB::table('tasks')
    ->join('user_tasks', 'tasks.id', '=', 'user_tasks.task_id')
    ->join('users','user_tasks.user_id', '=', 'user.id')
    ->select('users.name', 'tasks.title', 'tasks.start', 'tasks.end')
    ->get();
}
}

```

3.1.2 Разработка пользовательского интерфейса

Пользовательский интерфейс – это интерфейс, который обеспечивает передачу информации между пользователем и программно-аппаратными компонентами системы. Пользовательский интерфейс разрабатываемой системы состоит из компонентов: «Вход», «Регистрация» – для неавторизованного пользователя, и компонентов: «Задачи», «Проекты» для авторизованного пользователя.

Компонент Login (см. рис 3.1) содержит форму авторизации пользователя – поля email и пароль. Для этого использован специальный компонент библиотеки vuetify – v-form, внутри которого могут располагаться текстовые поля, чекбоксы, поля выбора и т.д. Аналогично компоненту Login реализован компонент регистрации (см. рис. 3.2).

Регистрация пользователей производится посредством электронной почты. После прохождения авторизации или регистрации пользователь попадает во вкладку «Задачи», в которой определены все текущие задачи пользователя по проектам, в которых он задействован (рис. 3.3).

Рисунок 3.1 – Форма авторизации

The image shows a registration form with a blue header containing the word 'Регистрация'. Below the header are three input fields: 'E-mail' with an envelope icon, 'Пароль' (Password) with a lock icon and a character count '0 / 15', and 'Повторить пароль' (Repeat password) with a lock icon and a character count '0 / 15'. At the bottom right of the form is a blue button with the text 'ЗАРЕГИСТРИРОВАТЬСЯ'.

Рисунок 3.2 – Форма регистрации

На вкладке «Задачи» отображаются задачи, назначенные текущему пользователю, то есть доступ к этим задач имеется только у него. Каждую задачу можно перемещать по стадиям, они будут перемещаться соответственно и в окне проекта, к которому эта задача относится. Форма добавления новой задачи изображена на рисунке 3.4, а сама карточка задачи на рисунке 3.5.

The screenshot shows a user interface for 'ЭлитСтройПроект-КС'. The top navigation bar includes 'ЗАДАНИЯ', 'ПРОЕКТЫ', 'ЛИЧНЫЙ КАБИНЕТ', and 'ВЫЙТИ'. Below this are four columns representing task stages: 'Задания' (Tasks), 'Выполняются' (In Progress), 'На рассмотрении' (Under Review), and 'Выполнено' (Completed). The 'Задания' column lists five tasks: 'Назначение', 'ТХ-часть', 'ПОС', 'ТХ-часть', and 'ПОС', all with dates '10.03.2020-15.04.2020'. The 'Выполняются' column shows one task 'Форма 10'. The 'На рассмотрении' column shows one task 'АР-часть'. The 'Выполнено' column shows one task 'Сметный расчет'.

Рисунок 3.3 – Задачи текущего пользователя

На рисунке 3.6 изображена карточка добавления нового проекта, а на рисунке 3.7 изображены все текущие проекты, в которых задействован пользователь

Новая задача

Название

Описание

Проект

Исполнитель

Стадия

Начало выполнения

Окончание

ПРИКРЕПИТЬ

СОЗДАТЬ

ОЧИСТИТЬ

ОТМЕНА

Рисунок 3.4 – Форма добавления новой задачи

Форма 10

10.03.2020-15.04.2020

Определить состав проекта "Урановый рудник"

ИЗМЕНИТЬ

СОХРАНИТЬ

Рисунок 3.5 – Карточка задачи

На рисунке 3.8 изображена карточка проекта с краткой информацией по проекту.

Новый проект

Название

Описание

Заказчик


ПРИКРЕПИТЬ 
 СОЗДАТЬ
 ОЧИСТИТЬ
 ОТМЕНА

Рисунок 3.6 – Создание нового проекта





Текущие проекты		+	
В процессе			
	Многоэтажное здание Jan 10, 2020 - Jan 20, 2020	ОТКРЫТЬ	ЗАКРЫТЬ
	Промышленное сооружение Jan 9, 2014 - Jan 20, 2020	ОТКРЫТЬ	ЗАКРЫТЬ
	Многоэтажное здание Jan 10, 2020 - Jan 20, 2020	ОТКРЫТЬ	ЗАКРЫТЬ
Закрыты			
	Урановый Рудник Jan 9, 2014 - Jan 20, 2020	ОТКРЫТЬ	

Рисунок 3.7 – Все проекты, в которых занята организация

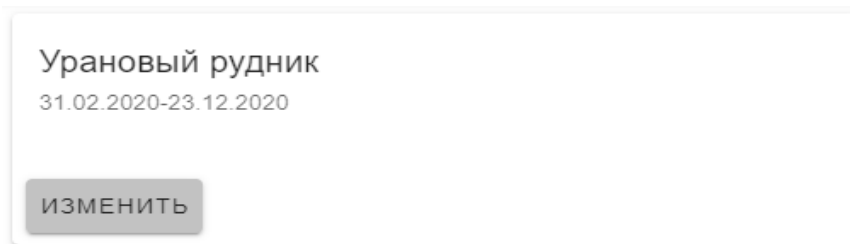


Рисунок 3.8 – Карточка проекта

3.1.3 Серверная часть

Написание серверной части предполагает написание маршрутизации и контроллеров приложения.

Маршрутизация — процесс определения лучшего пути, по которому пользователь может добраться до нужного ему компонента. В проектах, использующих Laravel и Vue используется два вида роутинга: роуты для шаблонов blade и vue-router.

В Laravel простейшие маршруты принимают URI (путь) и функцию-замыкание. На листинге 3.3 представлены маршруты роутов шаблонов blade.

Листинг 3.3 – Роуты шаблонов blade

```
Route::get('/', function () {  
    return view('welcome');  
});  
Auth::routes();  
Route::get('/home', 'HomeController@index')->name('home');
```

Для маршрутизации vue-компонентов используется плагин vue-router, маршруты прописываются в главном js файле – app.js (листинг 3.4)

Листинг 3.4 – app.js

```
window.Vue = require('vue');  
import VueRouter from 'vue-router'  
...  
Vue.use(VueRouter)  
...  
let routes = [  
    { path: '/tasks', component: require('./components/Tasks/Tasks.vue').default },  
    { path: '/projects', component: require('./components/Projects/Projects.vue').default },  
    { path: '/task/:id', component: require('./components/Tasks/Task.vue').default },  
    { path: '/project/:id', component: require('./components/Projects/Project.vue').default },  
    { path: '/addproject', component: require('./components/Projects/NewProject.vue').default },
```

```

        { path: '/addtask', component: require('./components/Tasks/NewTask.vue').de
fault }
    ]
    const router = new VueRouter({
        routes
    })
    const app = new Vue({
        el: '#app',
        vuetify: new Vuetify(),
        router,
        Vuedraggable
    });

```

Использование созданных роутов осуществляется следующим образом: в шаблоне `app.blade.php` добавляется скрипт `app.js` и ссылки на нужные компоненты, обернутые в `<router-link/>`. Для динамического отображения содержимого vue-компонентов используется `<router-view/>`.

Для создания заготовки системы аутентификации используется `php artisan make:auth`. В результате создаются шаблоны `blade`, защита роутов. Средства Laravel для аутентификации состоят из «защитников» и «провайдеров». Защитники определяют то, как аутентифицируются пользователи для каждого запроса. Например, Laravel поставляется с защитником «`session`», который поддерживает состояние с помощью хранилища сессий и `cookies`.

Провайдеры определяют то, как пользователи извлекаются из вашего постоянного хранилища. Laravel поставляется с поддержкой извлечения пользователей с помощью `Eloquent` и конструктора запросов БД.

На листинге 3.5 представлены контроллеры авторизации и регистрации.

Листинг 3.5 – Контроллеры регистрации и авторизации

```

class RegisterController extends Controller
{
    use RegistersUsers;
    protected $redirectTo = RouteServiceProvider::HOME; //после регистрации
пользователь попадет на домашнюю страницу приложения для авторизованных пользователей
    public function __construct()
    {
        $this->middleware('guest');
    }
    protected function validator(array $data) //валидация полей в форме регистрации
    {
        return Validator::make($data, [
            'name' => ['required', 'string', 'max:255'],

```

```

        'email' => ['required', 'string', 'email', 'max:255', 'unique:users'],
        'password' => ['required', 'string', 'min:8', 'confirmed'],
    ]);
}
protected function create(array $data)//функция создания нового пользователя и
внесение данных в БД после регистрации
{
    return User::create([
        'name' => $data['name'],
        'email' => $data['email'],
        'password' => Hash::make($data['password']),
    ]);
}}

```

Аналогичным образом создаются роуты и контроллеры для других элементов (задания, проекты и т.д.). Полный код представлен в приложении Б в листингах программ.

3.2 Контрольный пример

При регистрации нового пользователя данные вносятся в базу данных и пользователь переходит во вкладку «Задачи» (рис. 3.9 – 3.11).

The image shows a registration form with a blue header containing the word 'Регистрация'. Below the header are three input fields. The first field is labeled 'E-mail' and contains the text 'natdevelop21@gmail.com'. The second field is labeled 'Пароль' (Password) and contains six dots, with a '6 / 15' character count indicator to its right. The third field is labeled 'Повторить пароль' (Repeat password) and also contains six dots with a '6 / 15' character count indicator. At the bottom of the form is a blue button with the white text 'ЗАРЕГИСТРИРОВАТЬСЯ'.

Рисунок 3.9 – Регистрация нового пользователя

email	email_verified_at	password
natdevelop21@gmail.com	NULL	\$2y\$10\$B5b6SqBrpcdpxWv5bKPYoe8Fx5hTc7dOtTr3tYFtD2l...

Рисунок 3.10 – Запись в БД

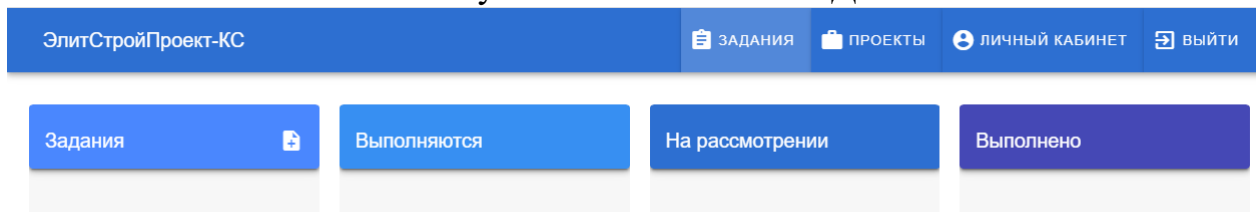


Рисунок 3.11 – Вкладка «Задания»

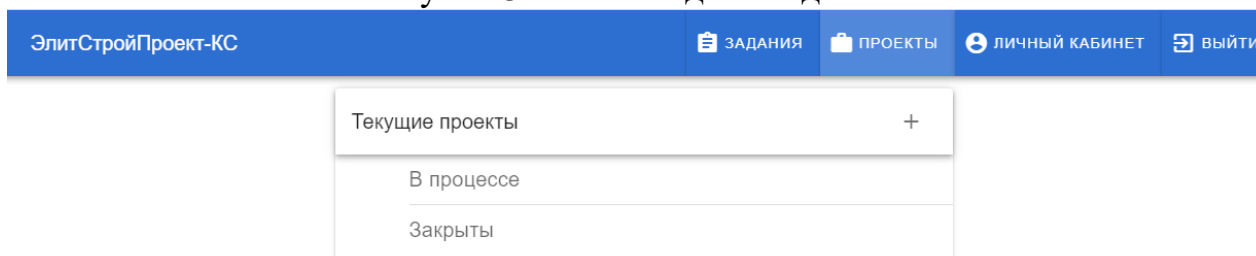


Рисунок 3.12 – Вкладка «Проекты»

Во вкладках «Задания» и «Проекты» пока что нет ни одной записи. Добавление нового проекта и внесение записи в БД показано на рисунках 3.13 – 3.15.

Новый проект

Название
Урановый рудник

Описание
Урановый рудник на месторождении Ирколь

Заказчик
Казатомпром

Начало выполнения
2020-04-05

ПРИКРЕПИТЬ СОЗДАТЬ

June 2021

S	M	T	W	T	F	S
		1	2	3	4	5
6	7	8	9	10	11	12
13	14	15	16	17	18	19
20	21	22	23	24	25	26
27	28	29	30			

ОТМЕНА ОК

Рисунок 3.13 – Заполнение формы создания нового проекта

+ Параметры							
	id	title	description	start	end	client	closed
	1	Урановый рудник	Урановый рудник на месторождении Ирколь	2020-04-05	2021-06-09	Казатомпром	0

Рисунок 3.14 – Запись нового проекта в БД

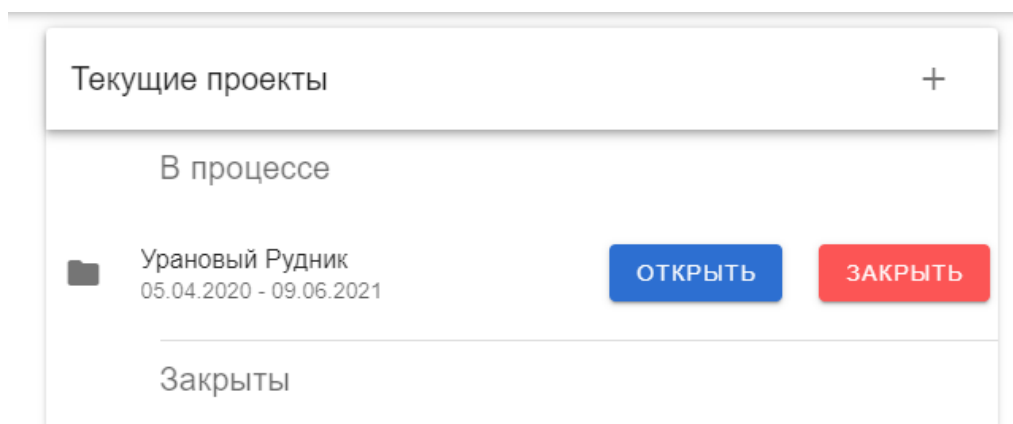


Рисунок 3.15 – Отображение новой записи во вкладке «Проекты»

Следующее на очереди добавление нового задания, для этого нужно перейти на вкладку «Задания» и нажать на иконку со знаком + в карточке «Задания». Скриншоты добавления нового задания показаны на рисунках 3.16 – 3.18.

Новая задача

Название
 Форма 10

Описание
 Сделать определение состава проекта
 Назначить сотрудников на задания

Проект
 Урановый рудник

Исполнитель
 natdevelop21@gmail.com

Начало выполнения
 2020-04-16

Окончание
 2020-04-22

ПРИКРЕПИТЬ
 СОЗДАТЬ
 ОЧИСТИТЬ
 ОТМЕНА

Рисунок 3.16 – Заполнение формы добавления нового задания

id	title	description	project_id	executor_id	start	end	stage
1	Форма 10	Сделать определение состава проекта Назначить сот...	1	1	2020-04-16	2020-04-22	1

Рисунок 3.17 – Добавление новой записи в БД

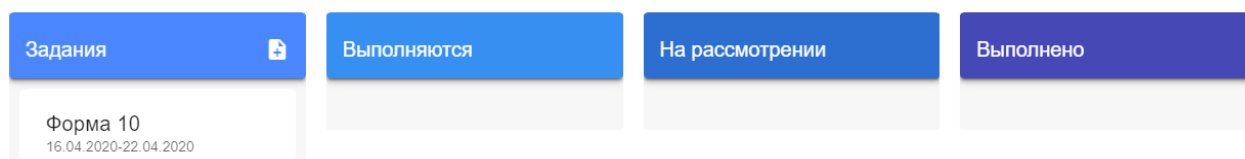


Рисунок 3.18 – Отображение новой записи во вкладке «Задания»

Добавленное задание можно перемещать по стадиям, в соответствии с ее выполнением. Допустим сейчас данное задание проверяется (рис. 3.19)

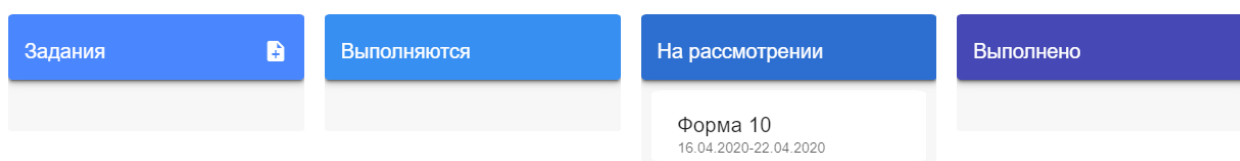


Рисунок 3.19 – Перемещение задания на другую стадию

В базе данных можно увидеть, что номер стадии у данного задания изменился (рис. 3.20).

id	title	description	project_id	executor_id	start	end	stage
1	Форма 10	Сделать определение состава проекта Назначить сот...	1	1	2020-04-16	2020-04-22	3

Рисунок 3.20 – Изменение стадии задания

Для большей наглядности создано большее количество пользователей, проектов и заданий (рис. 3.21 – 3.23).

id	title	description	start	end	client	closed
1	Урановый рудник	Урановый рудник на месторождении Ирколь	2020-04-05	2021-06-09	Казатомпром	0
2	Промышленное сооружение		2020-05-12	2020-10-19	Заказчик	0
3	Многоэтажное здание		2020-12-06	2021-10-04	Заказчик	0

Рисунок 3.21 – Новые проекты в базе данных

id	title	description	project_id	executor_id	start	end	stage
1	Форма 10	Сделать определение состава проекта Назначить сот...	1	1	2020-04-16	2020-04-22	3
2	ПОС	Проект организации строительства	1	1	2020-05-01	2020-05-21	1
3	КЖ-часть	Конструкции железобетонные	2	2	2020-05-20	2020-07-05	1
4	ПОС		3	2	2020-05-30	2020-07-25	1

Рисунок 3.22 – Новые задания в базе данных

id	name	email
1	Наталья	natdevelop21@gmail.com
2	Александр	alex@gmail.com

Рисунок 3.23 – Новые пользователи в базе данных

Тогда для пользователя Наталья вкладки «Задания» и «Проекты» будут выглядеть следующим образом (рис. 3.24 – 3.25).

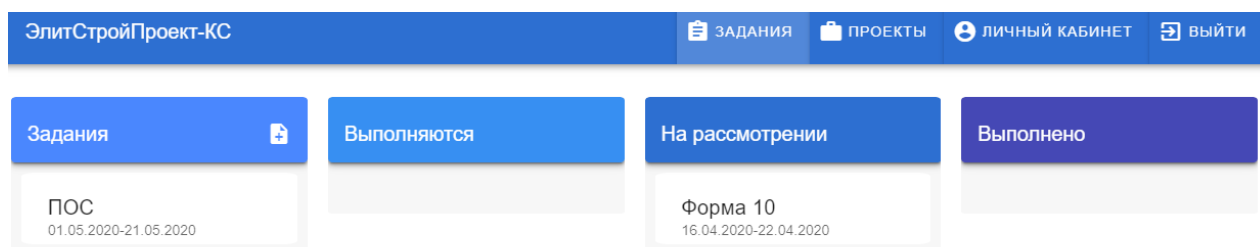


Рисунок 3.24 – Вкладка «Задания» пользователя Наталья

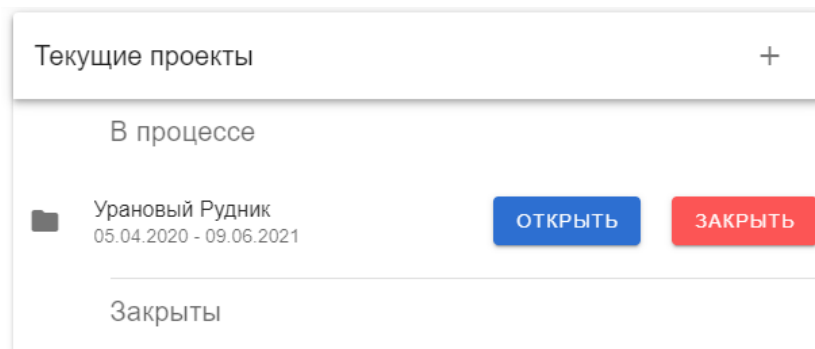


Рисунок 3.25 – Вкладка «Проекты» пользователя Наталья

Аналогично для пользователя Александр вкладки будут выглядеть следующим образом см. рис. 3.26-3.27.

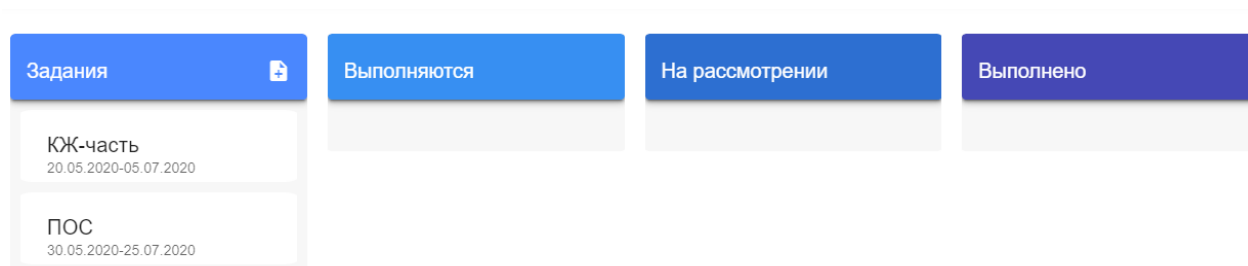


Рисунок 3.26 – Вкладка «Задания» пользователя Александр

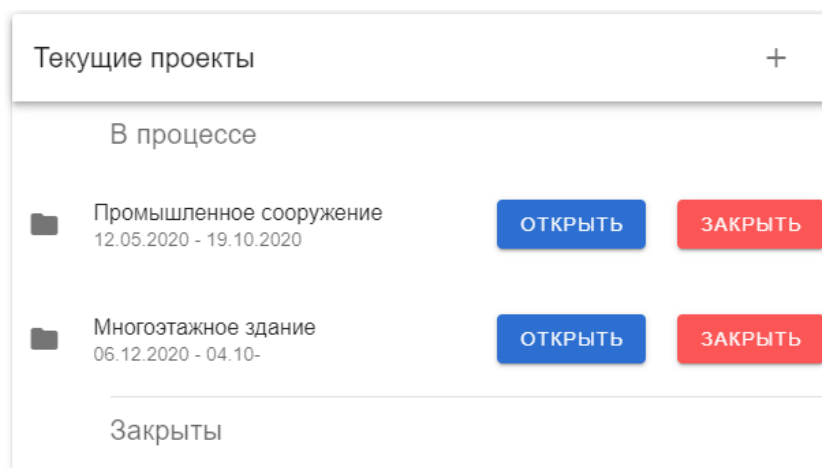


Рисунок 3.27–Вкладка «Проекты» пользователя Александр

Если проект уже выполнен или возникли другие причины, по которым его нужно закрыть, он переносится в раздел «Закреты» (рис. 3.28-3.30).

Закрытие проекта

Вы действительно хотите закрыть проект?

ОТМЕНА ДА

Рисунок 3.28 – Диалоговое окно о закрытии проекта

id	title	description	start	end	client	closed
1	Урановый рудник	Урановый рудник на месторождении Ирколь	2020-04-05	2021-06-09	Казатомпром	0
2	Промышленное сооружение		2020-05-12	2020-10-19	Заказчик	0
3	Многоэтажное здание		2020-12-06	2021-10-04	Заказчик	1

Рисунок 3.29 – Изменение значения атрибута «closed» для проекта

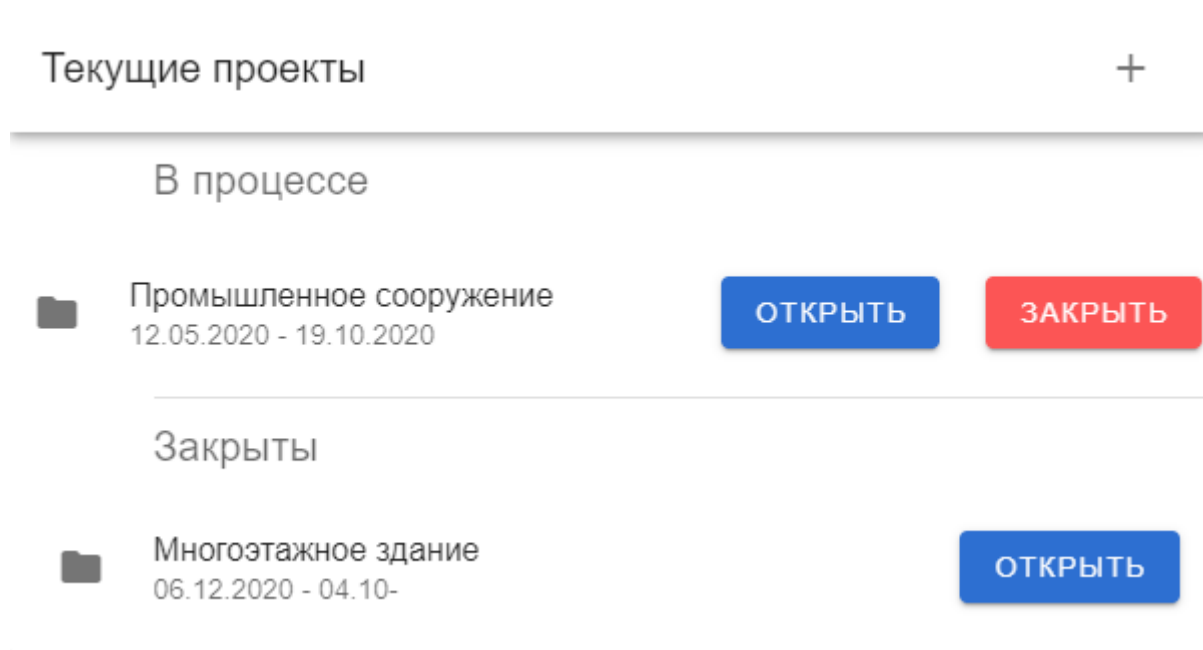


Рисунок 3.30 – Перенос проекта в раздел «Закреты»

Вывод. Таким образом, в ходе выполнения данного раздела дипломного проекта была произведена разработка пользовательского интерфейса, а также серверной части приложения. Была создана база данных, для связи с базой данных в Laravel проекте необходимо было сделать настройку БД в «.env» файле, а также создать модели и сделать миграцию. Для осуществления сложных запросов из нескольких таблиц баз данных, в том числе промежуточных, использовались конструкторы запросов.

4 Безопасность жизнедеятельности

4.1 Анализ условий труда при разработке ИС

Анализ условий труда в данной дипломной работе будет производиться для офиса, в котором происходила разработка приложения и где данный программный продукт будет использоваться.

Ежедневно офисные работники ощущают на себе влияние таких негативных факторов, как излучение монитора ПК, шум принтеров, сухой воздух, стресс, малоподвижный образ жизни[21].

На данном этапе, важно уточнить, что такое вредные и опасные факторы. Вредный производственный фактор – это фактор трудового процесса или среды, воздействие которого при определенных условиях на работника может вызвать профессиональное заболевание, снижение работоспособности. Опасный производственный фактор – фактор, способный стать причиной острого заболевания, резкого ухудшения здоровья или летального исхода. Говоря простым языком, вредные факторы действуют по принципу накопления и поражают работника в течении длительного времени, вызывают какое-либо заболевание. В свою очередь опасные факторы действуют, можно сказать, моментально, вызывают какую-либо травму.

В процессе работы на работников офиса оказывают неблагоприятное воздействие такие опасные и вредные производственные факторы, как:

- электромагнитное излучение;
- ионизирующее излучение;
- статическое электричество;
- напряженность электростатического поля;
- ионизация воздуха (повышенная или пониженная);
- повышенная яркость света, либо наоборот недостаток света;
- перегрузки костно-мышечного аппарата при долгом нахождении в сидячем положении;
- перегрузки мышц кистей рук, в процессе взаимодействия с клавиатурой и мышью;
- перенапряжение глаз;
- умственное перенапряжение;
- эмоциональное напряжение;
- монотонный труд.

Опасные и вредные факторы так же классифицируются на физические, химические, психофизические, биологические.

К физическим можно отнести такие факторы как высокие уровни рентгеновского, электромагнитного, инфракрасного, а также ультрафиолетового излучений. Помимо этого, к ним относятся повышенный уровень статического электричества и запыленности воздуха рабочей зоны, а также повышенное содержание положительно заряженных частиц и пониженное содержание отрицательно заряженных частиц, образующихся при расщеплении газовых молекул, в воздухе рабочей зоны. К тому же

повышенные уровни блескости и ослепленности, неравномерность распределения яркости в поле зрения, повышенная яркость светового изображения, повышенное значение напряжения в электрической цепи, замыкание которой может произойти через тело человека, повышенная или пониженная температура, влажность и подвижность воздуха, шум тоже относятся к физическим факторам[22].

К химическим факторам относятся повышенные содержания в воздухе рабочей зоны двуокиси углерода, озона, фенола, формальдегида. Наличие таких факторов напрямую связано с использованием недорогих синтетических материалов, чернил для принтеров, пластиковых панелей, каких-либо рабочих элементов оборудования и т.д. Все они выделяют такие токсичные вещества, как фенолы, бензол, формальдегид. Также при работе копировального или печатного оборудования выделяется озон, который ядовит в больших концентрациях[23].

Примерами психофизических вредных и опасных факторов могут послужить напряжение зрения и внимания, интеллектуальные, эмоциональные, длительные статические нагрузки, монотонность труда, необходимость обработки большого объема информации, нерациональная организация рабочего места. Причиной стрессов также могут послужить физические факторы, например шум, сквозняки и т.д.[23].

Биологические вредные и опасные факторы возникают при неверном расчете кратности воздухообмена, несвоевременной чистки систем и замены фильтрующих элементов кондиционеров и систем вентиляции, а также при неграмотном воздухообмене и несоблюдении правил уборки помещений. Такие факторы создают условия для развития патогенной микрофлоры, а также распространения бактерий, вирусов, спор грибков в потоке воздуха в помещении[23].

Разработка информационной системы по управлению проектированием производится по заказу проектно-строительной организации «ЭлитСтройПроект-КС».

ТОО «ЭлитСтройПроект-КС» находится по адресу Сатпаева 90/1. В разделе анализа и расчета будет рассматриваться кабинет, в котором производилась разработка приложения. Общий план офиса представлен на рисунке 4.1.

В кабинете расположено пять рабочих мест, четыре из которых оснащены персональными компьютерами с двумя и более мониторами, одно – ноутбуком ASUS. Данное оборудование не производит лишнего шума. В рассматриваемом помещении есть одно большое окно шириной 6 м, высотой 2,5 м, выходящее на восток. На против располагается здание торгового центра, этажность и высота, которого значительно уступают офисному зданию. В качестве источников искусственного освещения используются светильники Армстронг в количестве 6 штук, в каждом по 4 люминесцентные лампы. Отсюда можно сделать вывод, что проблем с шумом и освещением в данном помещении нет.

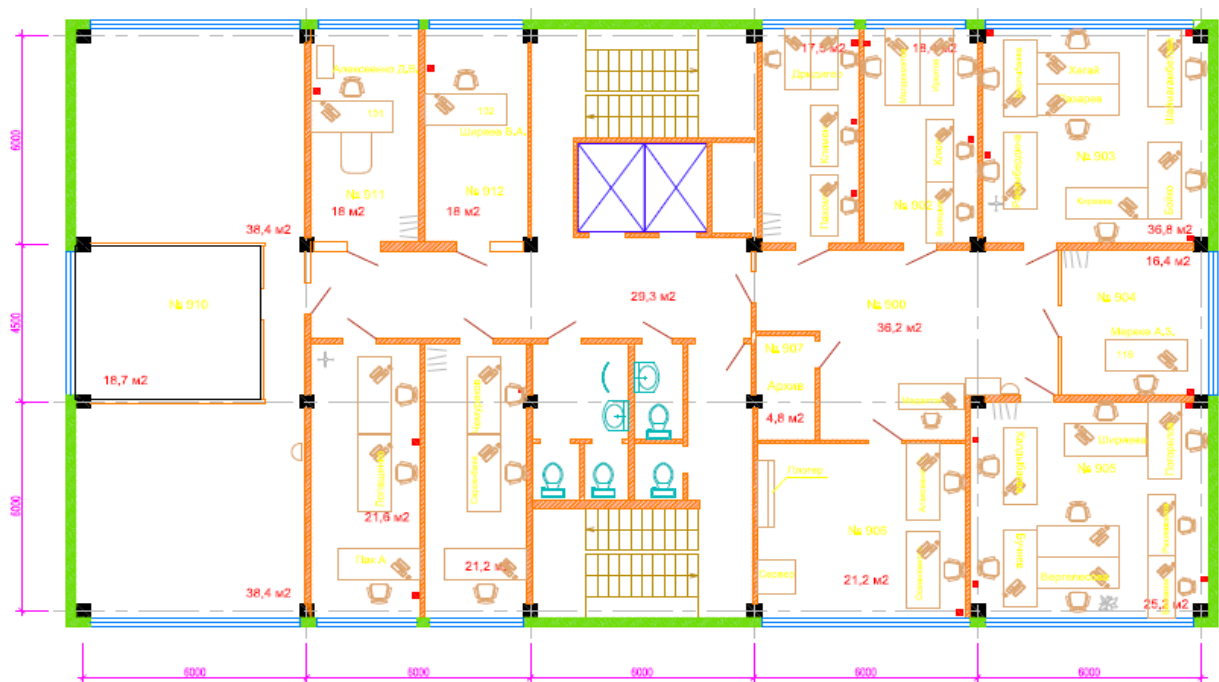


Рисунок 4.1 – План офиса

Но в кабинете, в котором происходит непосредственная разработка и эксплуатация информационной системы, нет кондиционера. Поэтому имеются проблемы с воздухообменом и температурой воздуха, особенно в весеннее и летнее время. В связи с этим необходимо произвести расчет вентиляционной системы и выбрать подходящую систему кондиционирования для кабинета, в котором ведется непосредственная разработка информационной системы.

4.2 Расчет аспирационной системы для производственного помещения

Таблица 4.1 – Исходные данные

Город	Алматы
Параметры помещения (Д*Ш*В), м	6,5 * 3,88 *4
Данные по оборудованию	
Количество компьютеров, шт.	5
Мощность $P_{об}$, кВт/ч	1,1
КПД η	0,8
Данные по источнику света	
Мощность $N_{осв}$, Вт/м ²	43
Вид источника освещения	люминесцентные лампы
Количество сотрудников	2 (мужчины), 3 (женщины)
Окна	
Количество, шт.	1
Площадь 1 окна, м ²	15
Расположение	ЮВ
Вид жалюзи	вертикальные металлические

Расчетное время суток, ч.	11-12
---------------------------	-------

Продолжение таблицы 4.1

Температура в помещении	
Летом, °С	26
Зимой, °С	20
Вид положения работы	сидячее

В данном расчете аспирационной системы будут рассчитываться тепловые нагрузки как внутренние, которые возникают внутри помещения, так и внешние (возникающие снаружи).

4.2.1 Расчет тепловых нагрузок снаружи помещения

Данные нагрузки представлены следующими составляющими:

- теплопоступления или теплопотери в результате разности температур снаружи и внутри здания через стены, потолки, полы, окна и двери;

- разность температур снаружи здания и внутри него летом является положительной, в результате чего имеет место приток тепла снаружи во внутрь помещения; и наоборот – зимой эта разность отрицательна и направление потока тепла меняется [12];

- теплопоступления от солнечного излучения через застекленные площади (данная нагрузка проявляется в форме ощущаемого тепла);

- теплопоступления от инфильтрации (проникновение воздуха с улицы через наружные ограждающие конструкции).

Теплопоступления и теплопотери в результате разности температур определяется по формуле (4.1):

$$Q_{огр} = V_{пом} * X_o * (t_{Нрасч} - t_{Врасч}) \quad (4.1)$$

где $V_{пом}$ – объем помещения (4.2), м³

$$V_{пом} = a * b * h = 6.5 * 3.88 * 4 = 100.88 \text{ м}^3 \quad (4.2)$$

X_o – удельная тепловая характеристика, Вт/м³ °С, $X_o = 0.42$;

$t_{Нрасч}$ – наружная температура. Для холодного периода – средняя температура самого холодного месяца в 13 часов, для теплого периода – средняя температура самого жаркого месяца в 13 часов [12].

$t_{Врасч}$ – внутренняя температура, выбирается с учетом комфортных условий или технологических требований, предъявляемых к производственным процессам [12].

Для теплого времени года (4.1):

$$t_{Нрасч} = 28 \text{ °С}$$

$$t_{Врасч} = 26 \text{ °С}$$

$$Q_{огр} = 100.88 * 0.42 * 2 = 84.74 \text{ Вт}$$

Для холодного времени года (4.1):

$$t_{\text{Нрасч}} = -17 \text{ }^\circ\text{C}$$

$$t_{\text{Врасч}} = 20 \text{ }^\circ\text{C}$$

$$Q_{\text{огр}} = 100.88 * 0.42 * 37 = 1567.68 \text{ Вт}$$

Избыточная теплота солнечного излучения в зависимости от типа стекла почти до 90% поглощается средой помещения, остальная часть отражается. Максимальная тепловая нагрузка достигается при максимальном уровне излучения, которое имеет прямую и рассеянную составляющие. Интенсивность излучения зависит от ширины местности, времени года и времени суток [12].

Теплопоступление от солнечного излучения через остекление определяется по формуле (4.3):

$$Q_P = (q^I F_O^I + q^{II} F_O^{II}) * \beta_{\text{с.з}} \quad (4.3)$$

где q^I, q^{II} – тепловые потоки от прямой и рассеянной солнечной радиации, Вт/м²;

F_O^I, F_O^{II} – площади светового проема, облучаемые и необлучаемые прямой солнечной радиации, м²;

$\beta_{\text{с.з}}$ – коэффициент теплопропускания. $\beta_{\text{с.з}} = 0.20$ [14].

При отсутствие наружных затеняющих козырьков, ребер и т. д. для периода облучения остекления солнцем, когда его лучи проникают через окно в помещение $F_O^I = F_O$; $F_O^{II} = 0$, (4.4):

$$Q_P = q^I F_O * \beta_{\text{с.з}} = (q_{\text{вп}} + q_{\text{вр}}) * K_1^C * K_2 * n * S_O \quad (4.4)$$

$q_{\text{вп}}, q_{\text{вр}}$ – тепловые потоки от прямой рассеянной радиации, Вт/м². Для широты в 44°СШ до полудня в 11-12 ч. По таблице 1 [12] для широты в 50°с.ш. до полудня в 11-12 ч. при расположении Ю: $q_{\text{вп}}, q_{\text{вр}}$

$$q_{\text{вп}} = 288 \text{ Вт/м}^2$$

$$q_{\text{вр}} = 85 \text{ Вт/м}^2$$

F_O – площадь светового проема (4.5) (n – число окон, S_O – площадь одного окна);

$$F_O = n * S_O = 1 * 15 = 15 \text{ м}^2 \quad (4.5)$$

K_1 – коэффициент затемнения остекления переплетами (K_1^C – для облученных проемов).

$$K_1^C = 0.72$$

K_2 – коэффициент загрязнения остекления.

$$K_2 = 0.9$$

Тогда по формуле (4.4):

$$Q_P = (288 + 85) * 0.72 * 0.9 * 0.2 * 15 = 725,11 \text{ Вт}$$

Тогда общее теплопоступление солнечного излучения равно:

$$Q_P = 725,11 \text{ Вт}$$

4.2.2 Расчет тепловых нагрузок внутри помещения

Внутренние тепловые нагрузки офисных помещениях складываются в основном из тепла:

- выделяемого людьми;
- выделяемого лампами и осветительными, электробытовыми приборами;
- выделяемого компьютерами, печатающими устройствами и фотокопировальными машинами и пр. (в офисных и других помещениях).

Теплопоступления от людей зависят от интенсивности выполняемой работы и параметров окружающего воздуха. Тепло, выделяемое человеком, складывается из ощутимого – передаваемого в воздух помещения путём конвекции и лучеиспусканий, и тепла, затрачиваемого на испарение влаги с поверхности кожи и из легких[12].

Летом при 26°C один мужчина выделяет явного тепла 61 Вт, а общего – 102 Вт. Женщина выделяет 85% от нормы тепловыделений взрослого мужчины[12]. Тогда выделение явного тепла в помещении составит:

$$Q_{\text{л}}^{\text{я}} = 61 * 2 + 61 * 3 * 0.85 = 277.55 \text{ Вт}$$

А выделение общего тепла:

$$Q_{\text{л}}^{\text{о}} = 102 * 2 + 102 * 3 * 0.85 = 464.1 \text{ Вт}$$

Зимой при 18°C один мужчина выделяет явного тепла 85 Вт, а общего – 105 Вт. Женщина выделяет 85% от нормы тепловыделений взрослого мужчины [12]. Тогда выделение явного тепла в помещении составит:

$$Q_{\text{л}}^{\text{я}} = 85 * 2 + 85 * 3 * 0.85 = 386.75 \text{ Вт}$$

Выделение общего тепла:

$$Q_{\text{л}}^{\text{о}} = 105 * 2 + 105 * 3 * 0.85 = 477.75 \text{ Вт}$$

Теплопоступление от осветительных приборов, оргтехники и оборудования рассчитывается следующим образом. Теплопоступление от ламп определяется по формуле (4.6):

$$Q_{\text{осв}} = \eta * N_{\text{осв}} * F_{\text{пол}} \quad (4.6)$$

где η – коэффициент перехода электрической энергии в тепловую (для люминесцентных ламп $\eta = 0.45 - 0.5$);

$N_{\text{осв}}$ – установленная мощность ламп ($N_{\text{осв}} = 43 \text{ Вт/м}^2$);

$F_{\text{пол}}$ – площадь пола ($F_{\text{пол}} = 6.5 * 3.88 = 25.22 \text{ м}^2$);

Тогда (4.6):

$$Q_{\text{осв}} = 0.45 * 43 * 25.22 = 488 \text{ Вт}$$

Тепло, выделяемое производственным оборудованием, определяется по формуле (4.7):

$$Q_{\text{об}} = N_{\text{уст}} * K, \quad (4.7)$$

где $N_{\text{уст}}$ – паспортная мощность оборудования ($N_{\text{уст}} = 1.1 \text{ кВт/ч}$);

K – единиц оборудования ($K = 5$);

Тогда (4.7):

$$Q_{\text{об}} = 1.1 * 5 = 5.5 \text{ кВт.}$$

Теплопритоки, возникающие за счет находящейся оргтехники – это 20% мощности оборудования (4.7):

$$Q_{об} = 1.1 * 5 * 0.2 = 1.1 \text{ кВт.}$$

4.2.3 Расчет теплового баланса помещения

На основании выполненных расчетов, баланс тепlopоступлений в помещении составит:

- летом:

$$Q_{изб} = 84,74 + 725,11 + 464,10 + 488 + 5500 + 1100 = 8361,95 \text{ Дж}$$

- зимой:

$$Q_{изб} = 1567,68 + 752,11 + 477,75 + 488 + 5500 + 1100 = 9885,54 \text{ Дж}$$

Так как тепловой баланс для зимы больше летнего теплового баланса, то рассчитаем тепло-напряжённость воздуха по формуле (4.8):

$$Q_H = \frac{Q_{изб} * 860}{V_{пом}} \quad (4.8)$$

$$Q_H = \frac{9885,54 * 860}{100,88} = 84\,274 \text{ ккал/м}^3 \approx 84,27 \text{ ккал/м}^3$$

При $Q_H > 20 \text{ ккал/м}^3$, $\Delta t = 8 \text{ }^\circ\text{C}$

Определение количества воздуха, необходимое для поступления в помещение (4.9):

$$L = \frac{Q_{изб} * 860}{C * \Delta t * \gamma} \quad (4.9)$$

где $C = 0.24 \text{ ккал/(кг}^\circ\text{C)}$ – теплоемкость воздуха,

$\gamma = 1.206 \text{ кг/м}^3$ – удельная масса приточного воздуха.

$$L = \frac{9885,54 * 860}{0.24 * 10^4 * 8 * 1.206} = 367.16 \frac{\text{м}^3}{\text{час}}$$

Определение кратности воздухообмена (4.10):

$$n = \frac{L}{V_{пом}} \quad (4.10)$$

$$n = \frac{367,16}{100,88} = 3,64 \text{ час}^{-1}$$

4.2.4 Выбор кондиционера

Исходя из полученных данных, выбран один кондиционер – FANTASIA FWFO-09HRN1. Характеристики этого кондиционера приведены в таблице 4.2.

Управление работой кондиционера производится с помощью пульта, который позволяет задать режим работы кондиционера. Существуют следующие режимы работы кондиционера: автоматический, ночной, обогрев, осушение воздуха, охлаждение, самодиагностика неисправностей, а также задать необходимую температуру, выбрать режим работы вентилятора.

Таблица 4.2 – Характеристики кондиционера

Тип	Настенная сплит-система
Обслуживаемая площадь	25 м ²

Мощность охлаждения	2630 Вт
Мощность обогрева	2780 Вт

Продолжение таблицы 4.2

Потребляемая мощность при охлаждении	770 Вт
Потребляемая мощность при обогреве	880 Вт
Размер (ШхВхГ)	715х285х194 мм
Поток воздуха (Выс/Ср/Низ)	403/320/292 м ³ /ч

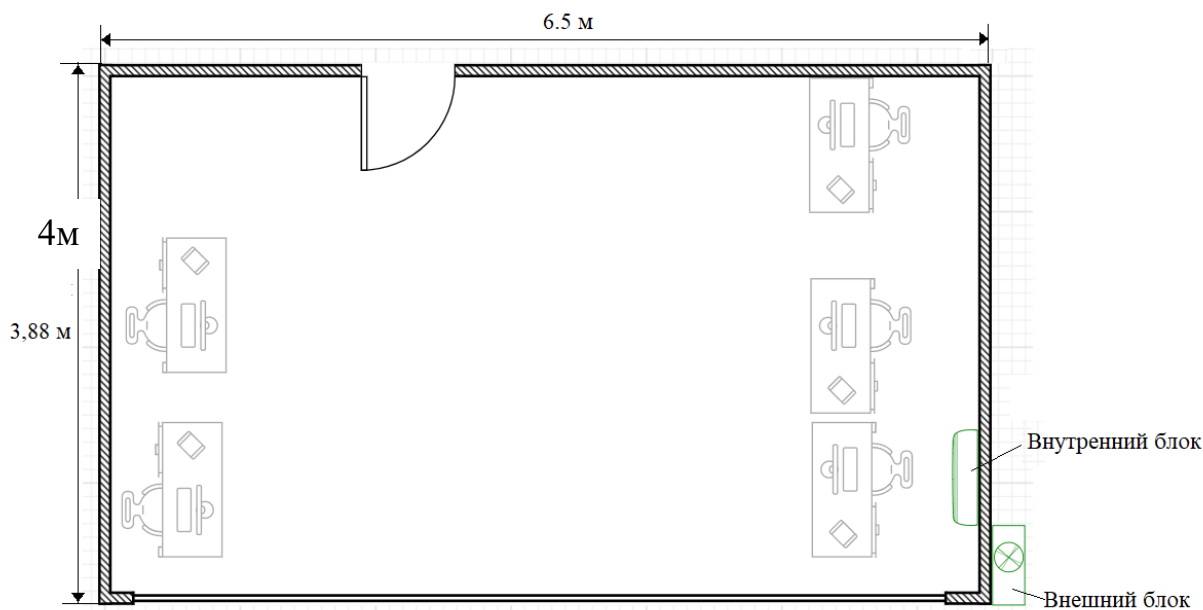


Рисунок 4.2 – Схема расположение кондиционера в производственном помещении

4.3 Эргономические требования к рабочему месту в офисе

Эргономические требования – это требования, которые предъявляются к системе ЧМС в целях оптимизации деятельности человека-оператора с учётом его социально-психологических, психофизических, психологических, антропометрических, физиологических и других объективных характеристик и возможностей [19].

В случае с рабочим местом в офисе, его главным атрибутом будет персональный компьютер. Организация рабочего места должна быть осуществлена таким образом, чтобы она обеспечивала: наилучшее размещение ПК и других инструментов, не допускала общего дискомфорта работника и повышала его продуктивность.

Общие принципы организации рабочего места[20]:

- на рабочем месте должны располагаться только необходимые для работы предметы и ничего лишнего;
- предметы, которые чаще используются находятся ближе, те что реже – дальше;
- рабочее место не должно быть загромождено;

- организация рабочего места должна обеспечивать необходимую обзорность.

Основные эргономические требования к рабочему месту за ПК изображены на рисунке 4.3.



Рисунок 4.3 – Размещение пользователя за рабочим столом

Эргономические требования к рабочему месту согласно санитарным правилам «Санитарно-эпидемиологические требования к условиям работы с источниками физических факторов (компьютеры и видеотерминалы), оказывающих воздействие на человека» [18]:

а) площадь на одно рабочее место пользователей ПК на базе плоских дискретных экранов при любом расположении – 4 м^2 . Площадь на одно рабочее место пользователей ноутбуков $2,5\text{ м}^2$;

б) расстановка компьютеров (ПК, ноутбуки) используется одним из трех 3-х вариантов: периметральная, рядные (2-3-рядная), центральная;

1) при периметральной расстановке, расстояние между стеной с оконными проемами и столами $0,5$ метров (далее – м), стеной и столами – $0,4\text{ м}$;

2) при рядной расстановке расстояние между тылом поверхности одного видеомонитора и экраном другого – не менее 2 м, между боковыми поверхностями видеомониторов не менее $1,2$ м, при двух-трехрядной расстановке одноместных столов с компьютерами расстояния в каждом ряду между боковыми поверхностями столов не менее $0,5$ м;

3) при центральной расстановке рабочие столы с компьютерами устанавливаются в центре, в два ряда без разрыва и экраны видеомониторов обращены в противоположные стороны, располагаясь в шахматном порядке, или напротив друг друга тыльными сторонами мониторов, при этом расстояние между тылом поверхности одного видеомонитора и экраном другого – не менее 2 м.

в) размеры рабочей поверхности:

1) высота рабочей поверхности стола (от пола) регулируется в пределах $640 - 800$ миллиметров (далее – мм);

2) ширину рабочей поверхности стола $800, 1000, 1200$ и 1400 мм;

3) рабочий стол имеет пространство для ног высотой не менее 580 мм, шириной – не менее 500 мм, глубиной – не менее 450 мм.

г) экран видеомонитора находится от глаз пользователя на расстоянии 600-700 мм, но не ближе 500 мм с учетом размеров алфавитно-цифровых знаков и символов;

д) продолжительность непосредственной работы с ПК и ноутбуками рекомендуется не более двух часов. В период работы проводятся профилактические мероприятия: упражнения для глаз через каждый 20-25 минут и физкультурная пауза через 45 минут во время перерыва.

В соответствии с вышеперечисленными эргономическими требованиями можно определить соответствует ли расстановка рабочих мест в офисе, площадь рабочего места и расстояния между ними требуемым.

Размеры офиса: длина – 6,5м, ширина 3,88м. Расстановка компьютеров периметральная (рис.4.4). Рабочие места выставлены по ширине, расстояние между рабочими местами и сами рабочие места не соответствуют эргономическим требованиям.

Для улучшения организации рабочих мест в офисе, необходимо изменить расстановку. Изменение расстановки рабочих мест показано на рисунке 4.5.

За счет перестановки рабочих мест были достигнуты требования к периметральной расстановке – расстояние между стеной с оконными проемами и столами 0,5 м, стеной и столами – 0,4 м

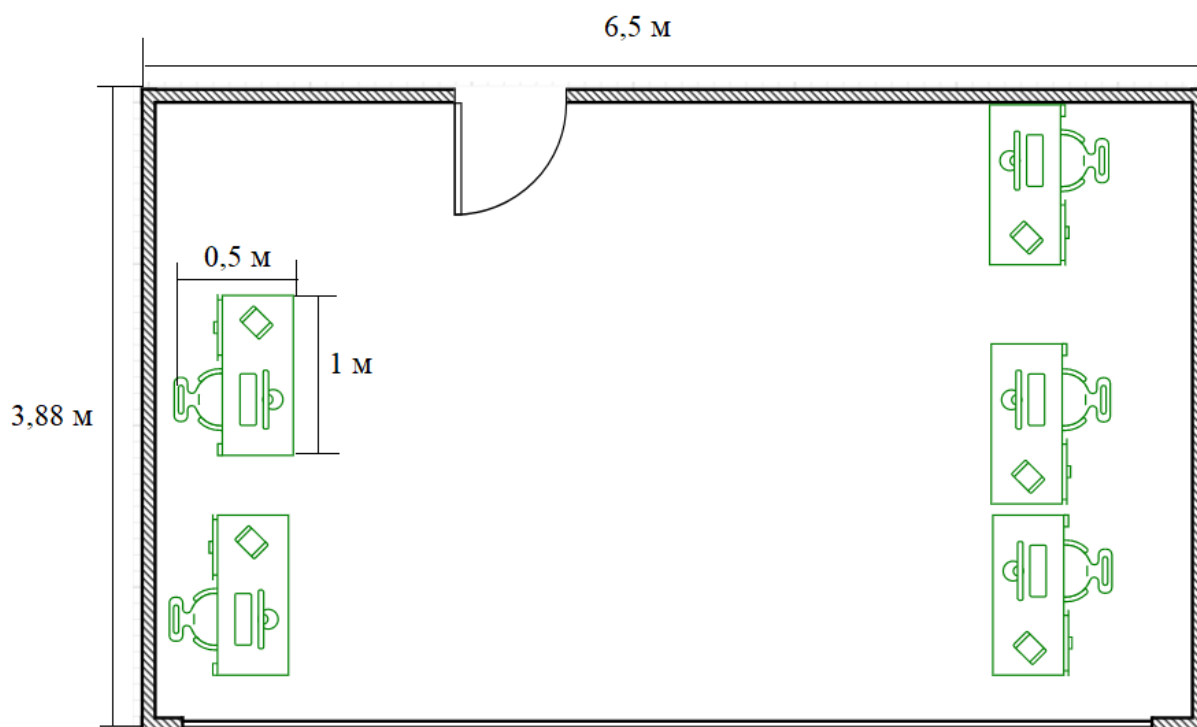


Рисунок 4.4 – План кабинета

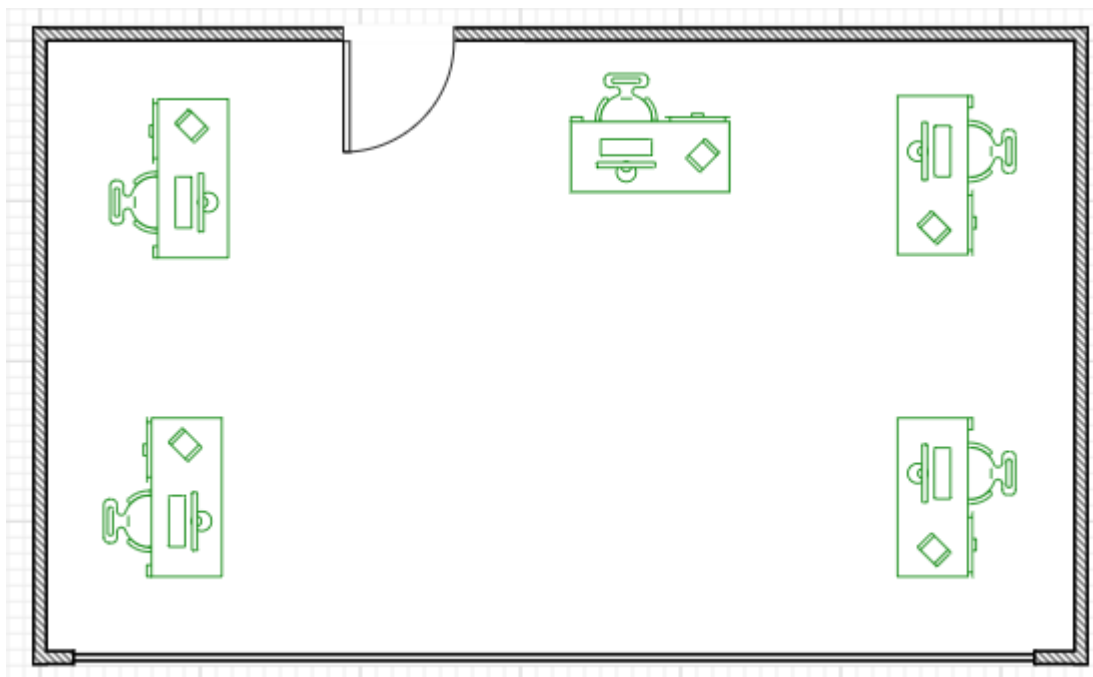


Рисунок 4.5 – Изменение расстановки рабочих мест в кабинете

4.4 Вывод

В части безопасности жизнедеятельности и охраны труда данного дипломного проекта был проведен анализ условий работы в офисе организации «ЭлитСтройПроект-КС». В ходе анализа были выявлены проблемы с системой кондиционирования в помещении, в котором происходила непосредственная разработка и написание дипломного проекта, а также с расположением рабочих мест в кабинете.

Для решения проблемы с кондиционированием был проведен расчет тепловых нагрузок внутри и снаружи помещения, расчет теплового баланса, по параметрам помещения и по рассчитанному количеству воздуха, необходимого для поступления в помещение был выбран кондиционер – FANTASIA FWFO-09HRN1. На рисунке 4.2 представлена схема его установки.

Расстановка рабочих мест в кабинете, а также их размеры не соответствуют санитарным требованиям работы с ПК. Для этого была произведена перестановка двух рабочих мест, за счет появилось требуемое расстояние между рабочими местами, а также между рабочими местами и стеной (рис. 4.5).

5. Экономическая часть

5.1 Резюме

В экономической части данного дипломного проекта будет приведено технико-экономическое обоснование разработки информационной системы для проектно-строительной организации. Данная информационная система предназначена для управления работой проектной группы организации. Основная цель создания такой информационной системы – упрощение процесса принятия решения по управлению проектированием, а также повышение эффективности работы сотрудников за счет тайм-менеджмента.

Пользователь приложения в лице сотрудника проектной группы может просматривать назначенные ему задачи, срок выполнения этих задач, а также приоритет этой задачи. Такой подход к управлению проектированием имеет преимущества, как со стороны руководства, так и проектировщиков, потому что наглядно отображает кто и чем занимается в определенный момент времени, показывает на какой стадии находится проект и позволяет в режиме реального времени обмениваться информацией и файлами по определенному проекту между сотрудниками, исключая взаимодействия посредством почты, сотовой связи, в письменном или устном виде.

ТЭО (технико-экономическое обоснование) служит для определения целесообразности создания продукта, в данном случае программного.

ТЭО состоит из анализа затрат на создание программного продукта, расчета цены самого программного продукта, эксплуатационных затрат, а также анализа и расчет результатов от создания и использования ПП (программного продукта).

5.2 Расчет трудоемкости разработки ИС

Базой для расчета плановой сметы затрат на разработку ПО является объем программного продукта. Общий объем (V_0) программного продукта определяется исходя из количества и объема функций, реализуемых программой[24]:

$$V_0 = \sum_{i=1}^n V_i, \quad (5.1)$$

где V_i – объем отдельной функции ПО;

n – общее число функций.

Оценивание объема программного продукта связано с выбором наиболее подходящей единицы измерения размера продукта. В дипломной работе в качестве единиц измерения объема ПО используется строка исходного кода (LOC)[24].

Расчет объема программного продукта (количества строк исходного кода) предполагает определение типа программного обеспечения (Приложение А [24]), всестороннее техническое обоснование функций ПО и определение объема каждой функции. На стадии технико-экономического обоснования проекта могут быть получены только ориентировочные

(прогнозные) оценки на основе имеющихся фактических данных по аналогичным проектам, выполненным ранее, или путем применения действующих нормативов (Приложение Б [24]). На основании информации о функциях разрабатываемого ПО по каталогу функций определяется объем функций и общий объем ПО, который уточняется (корректируется) с учетом условий разработки ПО в организации.

$$V_y = \sum_{i=1}^n V_{yi}, \quad (5.2)$$

где V_{yi} – уточненный объем отдельной функции ПО (LOC).

По уточненному объему ПО и нормативам затрат труда в расчете на единицу объема определяются нормативная и общая трудоемкость разработки ПО.

Для подсчета LOC использовалось расширение для среды разработки (IDE) WebStorm. В соответствии с ним, Проект состоит из 25 компонентов, общее число строк всех этих компонентов – 5129.

Нормативная трудоемкость разработки ПО (T_n) определяется на основании принятого к расчету объема (V_y) и категории сложности (Приложение В[24]), которая уточняется с учетом сложности и новизны проекта и степени использования стандартных модулей при разработке [24].

В соответствии с таблицей Г1 (Приложение Г [24]) разрабатываемая информационная система относится к 1 категории сложности, тогда нормативная трудоемкость её разработки составит $T_n=29$ чел./дн.

Коэффициент сложности рассчитывается по формуле

$$K_c = 1 + \sum_{i=1}^n K_i, \quad (5.3)$$

где K_i – коэффициент, соответствующий степени повышения сложности ПО за счет конкретной характеристики;

n – количество учитываемых характеристик.

Тогда, в соответствии с таблицей 1 [24], коэффициент сложности составит:

$$K_c = 1 + 0,06 + 0,07 + 0,12 = 1,25.$$

Поправочный коэффициент, учитывающий степень использования при разработке ПО стандартных модулей (K_m), определяется удельным весом этих модулей в общем объеме проектируемого продукта (см. таблицу 2 [24]) $K_m=0,8$.

Поправочный коэффициент, учитывающий новизну разрабатываемого ПО (K_n) определяется на основе данных таблицы 3 [24]. Разрабатываемая ИС относится к Б-категории новизны ПО и является развитием определенного параметрического ряда ПО, не используется на основе новых типов ПК и в средах новых ОС. Значение коэффициента новизны соответствующее этим факторам равно $K_n=0,9$.

Разработка информационной системы для управления проектированием реализуется одним исполнителем.

Процесс разработки данной ИС состоит из стадий разработки:

- техническое задание (ТЗ);

- эскизный проект (ЭП);
- технический проект (ТП);
- рабочий проект (РП);
- внедрение (ВН).

Для таких проектов рассчитывается общая трудоемкость с учетом стадий разработки:

$$T_0 = \sum_{i=1}^n T_i, \quad (5.4)$$

где T_i – трудоемкость разработки ПО на i -й стадии (чел./дн.);

n – количество стадий разработки.

Трудоёмкость стадий определяется на основе нормативной трудоемкости с учетом сложности, новизны и степени использования в разработке стандартных модулей ПО и удельного веса трудоемкости каждой стадии [24]:

$$T_{yi} = T_H \times d_{cmi} \times K_C \times K_m \times K_H \quad (5.5)$$

где T_{yi} – уточненная трудоемкость разработки ПО на i -й стадии;

T_H – нормативная трудоемкость;

d_{cmi} – удельный вес трудоемкости i -й стадии разработки ПО в общей трудоемкости разработки ПО (определяется по таблице 4 [24] в зависимости от степени новизны);

K_C – коэффициент, учитывающий сложность ПО, вводится на всех стадиях;

K_m – коэффициент, учитывающий степень использования стандартных модулей ПО, вводится только на стадии рабочего проекта;

K_H – коэффициент, учитывающий степень новизны ПО, вводится на всех стадиях.

Трудоемкость стадий ПО рассчитывается по следующим формулам [24]:

Трудоемкость стадии ТЗ:

$$T_{yз} = T_H \times K_C \times d_з \times K_H = 29 \cdot 1,25 \cdot 0,10 \cdot 0,9 = 3,26 \text{ чел/дн}$$

Трудоемкость стадии ЭП:

$$T_{yэ} = T_H \times K_C \times d_э \times K_H = 29 \cdot 1,25 \cdot 0,08 \cdot 0,9 = 2,61 \text{ чел/дн}$$

Трудоемкость стадии ТП:

$$T_{yп} = T_H \times K_C \times d_п \times K_H = 29 \cdot 1,25 \cdot 0,09 \cdot 0,9 = 2,93 \text{ чел/дн}$$

Трудоемкость стадии РП:

$$T_{yр} = T_H \times K_C \times d_р \times K_H \times K_m = 29 \cdot 1,25 \cdot 0,58 \cdot 0,9 \cdot 0,8 = 15,14 \text{ чел/дн}$$

Трудоемкость стадии ВН:

$$T_{yв} = T_H \times K_C \times d_в \times K_H = 29 \cdot 1,25 \cdot 0,15 \cdot 0,9 = 4,89 \text{ чел/дн}$$

Общая трудоемкость определяется как сумма трудоемкостей по стадиям:

$$T_y = T_{yз} + T_{yэ} + T_{yп} + T_{yр} + T_{yв} = 3,26 + 2,61 + 2,93 + 15,14 + 4,89 = 28,83 \text{ чел/дн}$$

Таблица 5.1 – Трудоемкость разработки программного продукта

Стадии разработки	Содержание работ	Трудоемкость чел/дн
1 Техническое задание	Постановка задачи.	3,26
	Сбор исходных материалов.	
	Выбор и обоснование критериев эффективности и качества разрабатываемой программы.	
	Определение требований к программе.	
	Разработка технико-экономического обоснования разработки программы.	
	Выбор языков программирования.	
	Согласование и утверждение технического задания	
2 Эскизный проект	Предварительная разработка структуры входных и выходных данных.	2,61
	Разработка макета ИС	
	Разработка общего описания алгоритма решения задачи.	
3 Технический проект	Разработка технико-экономического обоснования	2,93
	Разработка прототипа ИС	
	Определение формы представления входных и выходных данных.	
4 Рабочий проект	Разработка структуры программы.	15,14
	Программирование и отладка программы.	
5 Внедрение	Разработка, согласование и утверждение программы и методики испытаний.	4,89
	Оформление и утверждение акта о передаче программы на сопровождение и (или) изготовление.	
	Передача программы на сопровождение	
Итого		28,83

Трудоемкость в чел/ч при 8-часовом рабочем дне $T_y = 28,83 \cdot 8 = 230,64$ чел/ч.

5.3 Расчет затрат на разработку ИС

Расчет полных затрат на разработку проектного решения в виде информационных технологий (C_{ni}) осуществляется по формуле [24]:

$$C_{ni} = Z_{\text{фот}} + Z_{\text{сзи}} + M_i + P_{\text{си}} + P_{\text{ми}} + P_{\text{нки}} + P_{\text{зи}} + P_{\text{ни}} \quad (5.6)$$

где $Z_{\text{фот}}$ – общий фонд оплаты труда разработчиков, тенге;

$Z_{\text{сзи}}$ – отчисления по социальному налогу, тенге;

M_i – затраты на материалы, тенге;

$P_{\text{си}}$ – затраты на специальные программные средства, необходимые для разработки проектного решения, тенге;

$P_{\text{ми}}$ – затраты, связанные с эксплуатацией техники, тенге;

$P_{\text{нки}}$ – затраты на научные командировки, тенге;

$P_{\text{зи}}$ – прочие затраты, тенге;

$P_{\text{ни}}$ – накладные расходы, тенге.

Размер фонда оплаты труда разработчиков ($Z_{\text{фот}}$) рассчитывается по формуле:

$$Z_{\text{фот}} = Z_{oi} + Z_{di}, \quad (5.7)$$

где Z_{oi} – основная заработная плата, тенге;

Z_{di} – дополнительная заработная плата, тенге.

Основная заработная плата исполнителей по конкретному ПО рассчитывается по формуле:

$$Z_{oi} = \sum_{i=1}^n T_{ci} \times T_{\text{ч}} \times \Phi_{\text{п}} \times K, \quad (5.8)$$

где n – количество исполнителей, занятых разработкой конкретного ПО;

T_{ci} – часовая тарифная ставка i -го исполнителя (тенге/тыс. тенге);

$\Phi_{\text{п}}$ – плановый фонд рабочего времени i -го исполнителя (дней);

$T_{\text{ч}}$ – количество часов работы в день (час);

K – коэффициент премирования.

Базой для расчета основной заработной платы являются общая трудоемкость, плановая численность работников и плановые сроки разработки ПО [24].

По данным о специфике и сложности выполняемых функций составляется штатное расписание группы специалистов-исполнителей, участвующих в разработке ПО, с определением образования, специальности, квалификации и должности. Заработная плата full-stack разработчика согласно headhunter.kz начинается с 170 000 тенге в месяц при 8-часовом рабочем дне.

Таблица 5.2 – Сведения по работникам, задействованным в проекте

Специалист – Исполнитель	Количество, человек	Заработная плата в месяц, тенге
Программист	1	170 000
Итого		170 000

Часовая тарифная ставка рассчитывается путем деления месячной тарифной ставки на установленную при 40-часовой недельной норме рабочего времени расчетную среднемесячную норму рабочего времени в часах (Φ_p) [24]:

$$T_{\text{ч}} = \frac{T_{\text{м}}}{\Phi_p} = \frac{170000}{8 \cdot 21} = 1012 \text{ тг/ч} \quad (5.9)$$

где $T_{\text{ч}}$ – часовая тарифная ставка (тыс. тенге);

$T_{\text{м}}$ – месячная тарифная ставка (тыс. тенге).

Таблица 5.3 – Затраты на оплату труда

Категория работника	Трудоемкость разработки ПП, чел.×ч	Часовая ставка, тг/ч	Сумма, тг
Разработчик	230,64	1012	233 408
ИТОГО затраты на оплату труда			233 408

$Z_{oi} = 233\,408$ тенге.

Дополнительная заработная плата составляет в среднем 10% от основной заработной платы и рассчитывается по формуле:

$$Z_{di} = Z_{oi} \times N_d / 100 = 233\,408 \cdot 0,1 \approx 23\,341 \text{ тенге}$$

где N_d – коэффициент дополнительной заработной платы разработчиков.

$$Z_{\text{ФОТ}} = Z_{oi} + Z_{di} = 233\,408 + 23\,341 = 256\,749 \text{ тенге}$$

Социальный налог для юридических лиц рассчитывается:

$$Z_{\text{СН}} = (Z_{\text{ФОТ}} - \text{ОПВ} - \text{ВОСМС}) \cdot 0,095 - \text{СО} \quad (5.10)$$

где ОПВ – обязательный пенсионный взнос (10% от заработной платы);

ВОСМС – взнос на ОСМС (1% от заработной платы);

СО – социальные отчисления (3,5% от заработной платы с вычетом пенсионного взноса).

$$\text{ОПВ} = 256\,749 \cdot 0,1 = 25\,675$$

$$\text{ВОСМС} = 256\,749 \cdot 0,02 = 5\,135$$

$$\text{СО} = (256\,749 - 25\,675) \cdot 0,035 = 8\,088$$

$$\text{СН} = (256\,749 - 25\,675 - 5\,135) \cdot 0,095 - 8\,088 = 13\,376 \text{ тенге}$$

$$Z_{\text{сзи}} = \text{СН} + \text{СО} + \text{ВОСМС} = 13\,376 + 8\,088 + 5\,135 = 26\,599 \text{ тенге}$$

Величина затрат на материалы на основании исходных данных определяется по формуле:

$$M_i = \frac{Z_{\text{осн}} \cdot N_{\text{мз}}}{100} \quad (5.11)$$

где $N_{\text{мз}}$ – норма расхода материалов от основной заработной платы (3–5%).

Величина затрат на материалы при норме расхода – 3%:

$$M_i = \frac{233\,408 \cdot 3}{100} = 7\,002 \text{ тенге}$$

Расходы по статье «Спецоборудование» (P_{ci}) включают затраты средств на приобретение вспомогательных специального назначения технических и программных средств, необходимых для разработки конкретного ПО, включая расходы на их проектирование, изготовление, отладку, установку и эксплуатацию:

$$P_{ci} = \sum_{i=1}^n C_{ci} \quad (5.12)$$

где C_{ci} – стоимость конкретного специального оборудования (тыс. тенге);

n – количество применяемого специального оборудования. [24]

Таблица 5.4 – Затраты на спецоборудование

Наименование материального ресурса	Единица измерения	Количество израсходованного материала	Цена за единицу, тг	Сумма, тг
Ноутбук ASUS X542	шт	1	289 000	289 000
Мышь компьютерная Logitech M171	шт	1	8 000	8 000
Принтер лазерный Epson 1120	шт	1	52 000	52 000
ОС Windows 10 Pro (ключ активации)	шт	1	2 200	2 200
IDE WebStorm (лицензия на пользователя в месяц)	шт	1	5524	5524
Итого				356 724

Научные командировки не предусмотрены.

Расходы по статье «Прочие затраты» (P_{zi}) на конкретное ПО включают затраты на приобретение и подготовку специальной научно-технической информации и специальной литературы. Определяются по нормативу, разрабатываемому в целом по организации, в процентах к основной заработной плате [24]:

$$P_{zi} = Z_{oi} \cdot \frac{N_{пз}}{100} \quad (5.13)$$

где $N_{пз}$ – норматив прочих затрат в целом по организации в (%), в дипломной работе нужно брать 20%.

$$P_z = 233\,408 \cdot 0,2 = 46\,682 \text{ тенге}$$

Затраты по статье «Накладные расходы» (P_{ni}), связанные с необходимостью содержания аппарата управления, вспомогательных хозяйств

и опытных (экспериментальных) производств, а также с расходами на общехозяйственные нужды (P_{ni}), относятся на конкретное ПО по нормативу (H_{nr}) в процентном отношении к основной заработной плате исполнителей. Норматив устанавливается в целом по организации [24]:

$$P_{ni} = Z_{oi} \cdot \frac{H_{nr}}{100} \quad (5.14)$$

где P_{ni} – накладные расходы на конкретную ПО (тыс. тенге);

H_{nr} – норматив накладных расходов в целом по организации в (%), в дипломной работе нужно брать 70%.

$$P_{ni} = 233\,408 \cdot 0,7 = 163\,386 \text{ тенге}$$

Таблица 5.5 – Затраты на разработку

Затраты на разработку	Условное обозначение	Значение, тенге	В процентах от общей суммы
Фонд оплаты труда	Зфот	256 749	30%
Социальные отчисления и налоги	Зсзи	26 599	3,1%
Материалы	Ми	7702	0,9%
Спецоборудование	Рси	356 724	41,6%
Прочие затраты	Пзи	46 682	5,4%
Накладные расходы	Рни	163 386	19%
Итого	C_{ni}	857 842	100%

Рентабельность и прибыль по создаваемому ПО ($П_{oi}$) определяются исходя из результатов анализа рыночных условий, переговоров с заказчиком (потребителем) и согласования с ним отпускной цены, включающей дополнительно налог на добавленную стоимость. В случае разработки ПО для использования внутри организации оценка программного продукта производится по действующим правилам и показателям внутреннего хозрасчета (по ценам, устанавливаемым для расчета за услуги между подразделениями). Прибыль рассчитывается по формуле [24]:

$$П_{oi} = C_{ni} \cdot \frac{У_{рni}}{100}, \quad (5.15)$$

где $П_{oi}$ – прибыль от реализации ПО заказчику (тыс. тенге);

$У_{рni}$ – уровень рентабельности ПО (%), в дипломной работе брать 40-60%;

C_{ni} – себестоимость ПО (тыс. тенге).

$$П_{oi} = 857\,842 \cdot 0,4 = 343\,136 \text{ тенге}$$

Прогнозируемая цена ПО без налогов ($Ц_{pi}$):

$$Ц_{pi} = C_{ni} + П_{oi} = 857\,842 + 343\,136 = 1\,200\,978 \text{ тенге}$$

Прогнозируемая отпускная цена ($Ц_{oi}$):

$$Ц_{oi} = Ц_{pi} + \text{НДС} \quad (5.16)$$

Ставка налога на добавленную стоимость НДС в РК на 2020 год составляет 12% от отпускной цены ПО.

$$Ц_{01} = 1\,200\,978 + \frac{1\,200\,978 \cdot 12}{100} = 1\,345\,095 \text{ тенге}$$

Организация-разработчик участвует в освоении ПО и несет соответствующие затраты, на которые составляется смета, оплачиваемая заказчиком по договору. Затраты на освоение определяются по нормативу ($H_0=10\%$) от себестоимости ПО в расчете на 3 месяца и рассчитываются по формуле:

$$P_{0i} = C_{ni} \cdot \frac{H_0}{100} = 857\,842 \cdot 0,1 = 85\,784 \text{ тенге} \quad (5.17)$$

Затраты на сопровождение ПО (P_{Ci}). Организация-разработчик осуществляет сопровождение ПО и несет соответствующие расходы, которые оплачиваются заказчиком в соответствии с договором и сметой на сопровождение. Затраты на сопровождение определяются по установленному нормативу ($H_C=20\%$) от себестоимости ПО (в расчете на год) и рассчитываются по формуле [24]:

$$P_{Ci} = C_{ni} \cdot \frac{H_C}{100} = 857\,842 \cdot 0,2 = 171\,568 \text{ тенге} \quad (5.18)$$

Капиталовложения программного обеспечения с учетом затрат на освоение и сопровождение будет:

$$K = 1\,345\,095 + 85\,784 + 171\,568 = 1\,602\,447 \text{ тенге.}$$

5.4 Расчет эксплуатационных затрат

Годовые эксплуатационные текущие затраты в условиях функционирования информационных технологий (C) рассчитываются по формуле [24]:

$$C = ЗП + ОТ + А + М + НР \quad (5.19)$$

где ЗП—годовые затраты на оплату труда специалистов при выполнении ими своих функций в рамках автоматизируемого процесса после внедрения ИТ, тенге;

ОТ—отчисления по социальному налогу, тенге;

А—затраты на амортизацию, тенге;

М—годовые материальные затраты на сопровождение программного продукта, тенге;

НР—накладные расходы, тенге.

Годовые затраты по заработной плате специалистов после внедрения ИТ определяются по формуле:

$$ЗП = \frac{O_c \cdot Ч_c \cdot 12}{\Phi_{р.в.}} \cdot t_{общ} \cdot 12 \cdot (1 + K_d) \quad (5.20)$$

где O_c —оклад специалиста, тенге/мес.;

$Ч_c$ - численность специалистов, участвующих в процессе, чел.;

$\Phi_{р.в.}$ —годовой фонд рабочего времени, час;

$t_{общ}$ - трудоемкость решения задач в условиях функционирования ИТ в месяц, час;

K_d —коэффициент дополнительной заработной платы.

Трудоемкость решения задач в условиях функционирования ИТ в месяц вычисляется следующим образом [24]:

$$t_{\text{общ}} = \sum_{\beta=1}^n t_{\beta} \cdot K_{\beta}, \quad (5.21)$$

где t – затраты времени на решение β -й задачи, час:

K_{β} – количество решаемых β -х задач в месяц, ед.

В ходе эксплуатации системы предполагается появление серверных ошибок, на обработку одной ошибки в среднем уходит 2 часа, на протяжении месяца эксплуатации может возникать до 50 ошибок, помимо этого ожидается внесение изменений в интерфейс и доработка компонентов по требованию руководства. Всего компонентов в системе 25, на доработку одного уходит в среднем 5 часов.

$$t_{\text{общ}} = 2 \cdot 50 + 5 \cdot 25 = 100 + 125 = 225 \text{ часов}$$

$$\text{ЗП} = \frac{170\,000 \cdot 1 \cdot 12}{1968} \cdot 225 \cdot 12 \cdot 1 = 2\,798\,780 \text{ тенге}$$

Материальные затраты определяются по формуле:

$$M = \sum_{\mu=1}^n C_{\mu} \cdot N_{\mu} \quad (5.22)$$

где C_{μ} – цена μ -го вида единицы материальных затрат в условиях функционирования ИТ, тенге.

N_{μ} – используемое количество μ -го вида материальных затрат в месяц, ед.

В условиях функционирования разрабатываемой ИС, не предполагается материальных затрат.

Амортизационные отчисления производятся по установленным нормам амортизации, выражаются в процентах к балансовой стоимости оборудования и рассчитываются по формуле:

$$A = C_{\text{обор}} \cdot N_a \quad (5.23)$$

где N_a — норма амортизации (25 %);

$C_{\text{обор}}$ — первоначальная стоимость оборудования.

$$A = (289\,000 \cdot 0,25) + (8\,000 \cdot 0,25) + (52\,000 \cdot 0,25) = 87\,250$$

Отчисления по социальному налогу и накладные расходы рассчитываются так же, как и при разработке информационных технологий.

$$\text{СН} = (\text{ЗП} - \text{ОПВ} - \text{ВОСМС}) \cdot 0,095 - \text{СО}$$

$$\text{ОПВ} = 2\,798\,780 \cdot 0,1 = 279\,878 \text{ тенге}$$

$$\text{ВОСМС} = 2\,798\,780 \cdot 0,02 = 55\,975 \text{ тенге}$$

$$\text{СО} = (2\,798\,780 - 279\,878) \cdot 0,035 = 88\,162 \text{ тенге}$$

$$\text{СН} = (2\,798\,780 - 279\,878 - 55\,975) \cdot 0,095 - 88\,162 = 145\,816 \text{ тенге}$$

$$\text{ОТ} = \text{СН} + \text{СО} + \text{ВОСМС} = 145\,816 + 88\,162 + 55\,975 = 289\,953 \text{ тенге}$$

$$\text{НР} = \text{ЗП} \cdot \frac{N_{\text{НР}}}{100} = 2\,798\,780 \cdot 0,7 = 1\,952\,846 \text{ тенге}$$

Таблица 5.6 – Эксплуатационные затраты

Затраты на эксплуатацию	Условное обозначение	Значение, тенге	В процентах от общей суммы
Заработная плата	ЗП	2 798 780	54,6%
Социальные отчисления и налоги	ОТ	289 953	5,6%
Материальные затраты	М	0	0%
Амортизационные отчисления	А	87 250	1,7%
Накладные расходы	НР	1 952 846	38,1%
Итого		5 128 829	

$C = 5\,128\,829$ тенге

5.5 Расчет результатов от создания и использования ИС

Информационная система для управления проектированием предназначена в первую очередь для контроля работы проектной группы. Она упрощает процесс принятия решений по управлению, а также способна повысить эффективность и скорость работы сотрудников, за счет тайм-менеджмента.

Для оценки экономии от использования разрабатываемой ИС необходимо сравнить эксплуатационные расходы с ее применением и без.

Статьи затрат при применении ИС включают в себя:

- заработная плата специалиста, осуществляющего поддержку и сопровождение системы;
- износ оборудования;
- накладные расходы.

Данная система не предполагает расхода каких-либо материалов.

В организации имеется два куратора проектов (месячная заработная плата – 150 000 тенге), основной задачей которых является осуществление контроля работы проектной группы, управление проектированием.

$$ЗП = 3\,600\,000 \text{ тенге}$$

$$ОПВ = 3\,600\,000 \cdot 0,1 = 360\,000$$

$$СО = (3\,600\,000 - 360\,000) \cdot 0,035 = 113\,400$$

$$ВОСМС = 3\,600\,000 \cdot 0,02 = 72\,000$$

$$СН = (ЗП - ОПВ - ВОСМС) \cdot 0,095 - СО = 187\,560$$

$$ОТ = СО + СН + ВОСМС = 113\,400 + 72\,000 + 187\,560 = 372\,960 \text{ тенге}$$

В их распоряжении находятся два персональных компьютера (стоимость одного ~ 200 000 тенге), принтер (130 000 тенге) и прочая компьютерная периферия (~30 000 тенге); итого затрат на оборудование – 360 000 тенге.

Износ оборудования рассчитывается исходя из 25 % амортизационных отчислений за год.

$$A = (400\,000 \cdot 0,25) + (130\,000 \cdot 0,25) + (30\,000 \cdot 0,25) = 140\,000 \text{ тенге}$$

На заправку картриджей, бумагу прочую канцелярию в месяц уходит порядка 30 000 тенге.

Общие накладные расходы составят:

$$НР = ЗП \cdot \frac{N_{НР}}{100} = 3\,600\,000 \cdot 0,7 = 2\,520\,000 \text{ тенге}$$

Статьи затрат без применения ИС включают в себя:

- заработная плата специалистов – кураторов проектов, которые осуществляют управление проектами;
- износ используемого ими оборудования;
- расход материалов (к примеру канцелярия – бумага, картридж и т.д.)
- накладные расходы.

Таблица 5.7 – Годовые эксплуатационные затраты

Статьи	Без применения ИС (1 сотрудник – куратор проекта)	С применением ИС (1 сотрудник – программист, сопровождающий систему)
Годовая заработная плата	3 600 000	2 798 780
Социальные отчисления и налоги	372 960	289 953
Расходуемые материалы	360 000	0
Амортизационные отчисления	140 000	87 250
Накладные расходы	2 520 000	1 952 846
Всего	6 992 960	5 128 829

Ожидаемая условно-годовая экономия определяется по формуле [24]:

$$\mathcal{E}_{\text{уг}} = C_1 - C_2 + \sum \mathcal{E}_i \quad (5.24)$$

где $\mathcal{E}_{\text{уг}}$ - величина экономии, тенге;

C_1 и C_2 – показатели текущих затрат по базовому и внедряемому вариантам, тенге:

$\Sigma \Delta_i$ – ожидаемый дополнительный эффект от различных факторов, тенге.

$$\Delta_{\text{уг}} = C_1 - C_2 = 1\,864\,131 \text{ тенге.}$$

5.6 Расчет основных показателей экономической эффективности

Так как разработанная информационная система несет более социальный эффект, чем экономический, целесообразно оценивать его эффективность за счет экономии в сравнении с предыдущим периодом работы без использования ИС.

Величина ожидаемого годового экономического эффекта от внедрения ИС рассчитывается по формуле[24]:

$$\Delta_{\text{г}} = \Delta_{\text{уг}} - K \cdot E_{\text{н}} \quad (5.25)$$

где $\Delta_{\text{г}}$ – ожидаемый годовой экономический эффект, тенге;

$\Delta_{\text{уг}}$ – ожидаемая условно-годовая экономия, тенге;

K – капитальные вложения, тенге;

$E_{\text{н}}$ – нормативный коэффициент экономической эффективности капитальных вложений.

Нормативный коэффициент экономической эффективности капитальных вложений определяется по формуле:

$$E_{\text{н}} = \frac{1}{T_{\text{н}}} \quad (5.26)$$

где $T_{\text{н}}$ – нормативный срок окупаемости капитальных вложений, лет.

Нормативный срок окупаемости капитальных вложений принимается исходя из срока морального старения -технических средств и проектных решений ИС ($T_{\text{н}}=1,2,3\dots n$), для программных продуктов срок окупаемости принимаем равным 4 года.

$$E_{\text{н}} = \frac{1}{4} = 0,25$$

$$\Delta_{\text{г}} = 1\,864\,131 - 1\,602\,447 \cdot 0,25 = 1\,463\,519 \text{ тенге}$$

Расчетный коэффициент экономической эффективности капитальных вложений составляет:

$$E_{\text{р}} = \frac{\Delta_{\text{уг}}}{K} \quad (5.27)$$

где $E_{\text{р}}$ – расчетный коэффициент экономической эффективности капитальных вложений;

$\Delta_{\text{уг}}$ – ожидаемая условно-годовая экономия, тенге;

K – капитальные вложения на создание системы, тенге.

$$E_{\text{р}} = \frac{1\,864\,131}{1\,602\,447} = 1,16$$

Расчетный срок окупаемости капитальных вложений составляет:

$$T_{\text{р}} = \frac{1}{E_{\text{р}}}, \quad (5.28)$$

где E_p – коэффициент экономической эффективности капитальных вложений.

$$T_p = \frac{1}{1,16} = 0,86 \text{ года} \approx 10,3 \text{ месяцев}$$

Таблица 5.8 – Показатели сравнительной экономической эффективности от внедрения информационной системы

Наименование показателей	Значение
Условная годовая экономия затрат, тенге	1 864 131
Коэффициент экономической эффективности капитальных вложений (E_p)	1,16
Срок окупаемости капитальных вложений (T_p)	0,86

Таким образом, разработанная информационная система позволила не только упростить процесс управления работой проектной группы организации и процесс принятия решений руководством, но и значительно сэкономить на различных факторах. Во-первых, автоматизация процесса управления позволит больше не использовать услуги дополнительных кураторов проектов по филиалам, так как теперь отсутствует необходимость осуществлять контроль непосредственно на месте филиала, вся работа организации сосредоточена в ИС. Во-вторых, значительно снизится расход материалов, за счет того, что назначение на задания и все изменения по заданиям можно сделать напрямую в системе, не прибегая к формированию документов по назначению и их распечатке.

Ожидаемый годовой экономический эффект составил 1 463 519 тенге. Приложение окупится в первые 10 месяцев использования.

Заключение

Результатом выполнения данного дипломного проекта является система по управлению работой проектной группы в организации «ЭлитСтройПроект-КС». Данная система реализована в виде кроссплатформенного веб-приложения с применением новейших технологий в этой сфере. Это веб-приложение позволило организации автоматизировать процесс управления работой проектной группы и процесс принятия решений по проектированию.

В ходе выполнения дипломного проекта были достигнуты следующие цели:

- был проведен анализ процесса управления работой проектировщиков, а также были выявлено отсутствие определенной методики при управлении;
- изучены методики управления проектированием и выбраны наиболее подходящие под нужды организации;
- сделан сравнительный анализ аналогичных систем по управлению проектированием;
- определены основные требования и функционал системы;
- спроектирована база данных;
- разработка клиентской и серверной частей системы;
- произведены экономические расчеты для определения окупаемости приложения, в соответствии с этими расчетами разработанное веб-приложение экономит до 1 864 131 тенге в год и окупится в первые 10 месяцев его применения;
- сделан анализ условий труда в офисе организации, в ходе которого были выявлены проблемы с системой кондиционирования в помещении, а также проблемы с расположением рабочих мест сотрудников, для решения этих проблем спроектирована новая система кондиционирования, а также произведена перестановка рабочих мест в кабинете, для обеспечения более комфортной работы сотрудников.

В процессе выполнения дипломного проекта был проведен анализ процесса управления работой проектной группы, а также анализ работы самой проектной группы. В результате этого анализа были получены ценные знания в области, для которой разрабатывалась система. Данное веб-приложение с учетом специфики работы проектной группы позволяет упростить управление работой, а также процесс предоставления отчетности по выполненной работе. Информацию о выполненных задачах можно просмотреть в любое время, для этого требуется лишь доступ в интернет.

Данный продукт в значительной степени упрощает процесс организации работы, поскольку все назначения и разъяснения по задачам отображаются сразу в приложении. Именно такой подход позволяет одновременно упростить процесс предоставления отчетности для сотрудников и процесс управления и принятия решений по управлению для руководства организации. Помимо этого, данное веб-приложение позволит повысить

скорость работы сотрудников, а также эффективность за счет менеджмента своего рабочего времени.

В ходе написания данного дипломного проекта были закреплены знания, полученные в период теоретического обучения, а именно знания в областях разработки веб-приложений, объектно-ориентированного программирования, реляционных баз данных. К тому же были изучены современные методы управления проектированием, процесс разработки веб-приложения, архитектуры приложений и их виды.

В настоящее время одностраничные веб-приложение – SPA (Single Page Application) набирают большую популярность из-за своей кроссплатформенности и адаптивности. Поэтому реализация системы именно в таком формате наиболее актуальна для организации. Помимо этого, нужно отметить, что правильная организация управления рабочим процессом в большинстве случаев приводит к повышению скорости и эффективности работы сотрудников. Что подтверждает актуальность данного приложения.

Список литературы

1 Matthew Guay. Project Management 101: The Complete Guide to Agile, Kanban, Scrum and Beyond // ZAPIER.COM: Zapier team eBook, 2016. URL: <https://zapier.com/learn/project-management/project-management-systems/> (дата обращения: 17.01.2020).

2 Хабаров В. Особенности выбора методов управления проектом / Хабаров В.И., Тищенко К.В. – М: Московская академия предпринимательства, 2017 – 58с.

3 Родионова О. Преимущества использования гибких методологий разработки программного обеспечения / Родионова О.И., Ушачёв А.А. – Приамурский государственный университет им. Шолом-Алейхема, 2018. – 73с.

4 ДеМарко Т. Человеческий фактор. Успешные проекты и команды [Текст] / ДеМарко Т., Листер Т. – СПб: Символ-Плюс, 2011. – 256с.

5 ДеМарко Т. Deadline / ДеМарко Т. – Москва: Манн, Иванов, Фербер, 2014. – 352с.

6 Сазерленд Д. Scrum. Революционный метод управления проектами [Текст] / Сазерленд Д. – Москва: Манн, Иванов, Фербер, 2015. – 320с.

7 Степанов А. Сервисы и системы управления проектами // BLOG.GANTTPRO: Блог по проект-менеджменту. 2019. URL: <https://blog.ganttpro.com/ru/sistemy-instrumenty-servisy-upravlenie-proektami/> (дата обращения: 12.02.2020)

8 Телижняк Ю. Разбираемся с SPA и PWA: основные отличия и оптимизация динамических сайтов и одностраничных приложений // WEBPROMOEXPERTS.NET: Академия интернет-маркетинга. 2019. URL: <https://webpromoeexperts.net/blog/razbiraemsa-s-spa-i-pwa/> (дата обращения: 29.02.2020)

9 Сравнительный анализ системных характеристик СУБД Microsoft SQL Server, MySQL и PostgreSQL // DB-ENGINES.COM: инициатива по сбору и представлению информации о СУБД. 2020. URL: <https://dbengines.com/en/Microsoft+SQL+Server3%BMysql3%BPostgreSQL/> (дата обращения: 10.03.2020)

10 Постановка задач на разработку информационных систем // ONLINE.ZAKON.KZ: Информационная система «Параграф». URL: https://online.zakon.kz/Document/?doc_id=33787937#pos=3;-135/ (дата обращения: 12.03.2020)

11 Никитина Н. Дистанционный курс "Информационные Системы" для студентов направления подготовки "педагогическое образование" / Никитина Н.О., Табачук Н.П. – Тихоокеанский государственный университет, 2016 – 530с.

12 Хакимжанов Т.Е. Расчет аспирационных систем. Дипломное проектирование. Для студентов всех форм обучения всех специальностей. – Алматы: АИЭС, 2002. – 30 с.

13 СНиП РК 2.04-01-2001. Общие строительные нормы и правила устройства систем вентиляции и кондиционирования воздуха. Административные и бытовые здания.

14 Коэффициенты теплопропускания солнцезащитных устройств // BASE.GARANT.RU: Полные тексты документов в последней редакции. URL:<http://base.garant.ru/2306019/38d0e20d10a9099ed1e190abf152a12a/> (дата обращения: 16.04.2020)

15 Пособие 2.91 к СНиП 2.04.05-91 «Расчет поступления теплоты солнечной радиации в помещения»

16 Статья «Поступление теплоты в помещения гражданских и производственных зданий» на сайте // HVACSCHOOL.RU: Учебно-консультационный центр Университета Климата. URL:https://www.hvacschool.ru/biblioteka/tepl_balans/metodika_rascheta/teploti_omeshhen/(дата обращения: 16.04.2020)

17 Кондиционер FANTASIA // ACLIMAT.KZ: кондиционеры в Алматы. URL:<https://aclimat.kz/kondicioneryi/nastennyie/fwfo-09hrn1-s/>(дата обращения: 16.04.2020)

18 Санитарные правила «Санитарно-эпидемиологические требования к условиям работы с источниками физических факторов (компьютеры и видеотерминалы), оказывающих воздействие на человека» от 21 января 2016 года №38

19 Бударин Е. Особенности принципа эргономичности в архитектуре и дизайне современного жилища / Бударин Е.Л., Сапрыкина Н.А. – Самара: Предприятие "Новая техника", 2016 – 215с.

20 Эргономические требования к рабочему месту в офисе // STUDIZBA.COM: Эргономика для дизайна интерьера. URL:<https://studizba.com/lectures/8-dizayn/244-ergonomika-dlya-dizaynerov-interera/2976-2-ergonomicheskie-trebovaniya.html/> (дата обращения: 18.04.2020)

21 Негативные факторы воздействия в офисе // 34.ROSPOTREBNADZOR.RU: управление федеральной службы по надзору в сфере защиты прав потребителей и благополучия человека. 2011. URL:<http://34.rospotrebnadzor.ru/content/193/5757/>(дата обращения: 18.04.2020)

22 ГОСТ 12.0.003-2015 Система стандартов безопасности труда (ССБТ). Опасные и вредные производственные факторы. Классификация // DOCS.CNTD.RU: Электронный фонд правовой и нормативно-технической документации. URL: <http://docs.cntd.ru/document/1200136071> (дата обращения: 18.04.2020)

23 Вредные факторы в офисных помещениях // TESTSLAB.RU: лабораторные исследования. 2015. URL: <https://testslab.ru/stati/vrednye-factory-ofisnyh-pomeshchenij/> (дата обращения: 19.04.2020)

24 Г. Боканова Методические указания по выполнению экономической части дипломных работ Алматы, АУЭС, 2020 – 35с.

25 Техничко-экономическое обоснование дипломных проектов Брест, БГТу, 2014 – 15с.

26 Методические указания к выполнению расчетно-графической работы для студентов специальности 080801 «Прикладная информатика (по областям)» Уфа, 2010 – 12с.

27 Методические указания по выполнению экономической части дипломных работ Москва, Московский университет им. С.Ю. Витте, 2016 – 21с.

28 Симионов Ю.Ф., Боромотов В.В. Информационный менеджмент. — Ростов н.Д: Феникс, 2013, 250с.

Приложение А

Техническое задание

Техническое задание на разработку информационной системы для управления проектированием

1 Общие требования:

- наименование разрабатываемой системы:

Информационная система для управления проектированием «Elite» (ИСУП Elite);

- цели разработки:

а) автоматизация процессов управления проектно-строительной организацией;

б) обеспечение сбора и первичной обработки исходной информации, необходимой для подготовки отчетности по показателям деятельности;

в) создание единой системы отчетности по показателям деятельности;

г) повышения скорости и качества работы сотрудников.

- характеристика объектов автоматизации:

Основной объект автоматизации – строительные проекты, а точнее информация о них, которая бы позволила получить полное представление о состоянии работ по этому проекту. Результат автоматизации должен представлять собой краткую, но емкую информацию о проекте, которая позволит судить о том, на каком этапе находится проект, кто из сотрудников работает над проектом и насколько качественно.

2 Технические требования:

- требования к структуре и функционированию системы:

а) ИС должна быть централизованной, т.е. все данные должны располагаться в центральном хранилище;

б) веб-интерфейс для back-end должен работать на всех современных браузерах;

г) использование куки-файлов пользователя для выявления наиболее частых действия пользователя при работе с веб-приложением;

д) обеспечение защиты данных пользователей.

- требования к функциям, выполняемым системой:

а) просмотр информации по назначаемым задачам и проектам;

б) добавление новых проектов и задач;

г) изменение текущих проектов и задач по ним;

д) архивирование проектов;

е) удаление проектов и задач по ним.

- требования к интерфейсу системы:

а) окна регистрации, авторизации, просмотра, добавления задач и проектов;

б) удобство, интуитивная понятность.

Приложение Б

Листинг программы

Task.vue

```
<script>
export default {
  computed: {
    blocks () { return this.$store.getters.blocks }
  },
  methods: {
    updateBlock (id, status) {this.blocks.find(b => b.id === Number(id)).status = status } }
}
</script>
```

NewTask.vue

```
<script>
export default {
  data () {
    return {
      titleRules: [
        v => !!v || 'Введите название'
      ],
      projectRules: [
        v => !!v || 'Выберите проект'
      ],
      executorRules: [
        v => !!v || 'Выберите исполнителя'
      ],
      deadlineRules: [
        v => !!v || 'Выберите сроки'
      ]
    }
  },
  computed: {
    loading () {
      return this.$store.getters.loading
    }
  },
  methods: {
    createTask () {
```

Продолжение приложения Б

```
if (this.$refs.form.validate()) {
  const task = {
    title: this.title,
    description: this.description,
    start: this.start,
    end: this.end,
    project: this.project,
    stage: this.stage
  }
  this.$store.dispatch('createTask', task)
  .then(() => {
    this.$router.push('/tasks')
  })
  .catch(() => {})
}
},
clear () {
  this.title = ""
  this.description = ""
  this.project = null
  this.executor = null
  this.start = ""
  this.end = ""
  this.startMenu = false
  this.endMenu = false
}
}
}
</script>
```

App.vue

```
<script>
export default {
  data () {
    return {
      drawer: false
    }
  },
```

Продолжение приложения Б

```
computed: {
  error () {
    return this.$store.getters.error
  },
  isUserLoggedIn () {
    return this.$store.getters.isUserLoggedIn
  },
  links () {
    if (this.isUserLoggedIn) {
      return [
        {title: 'Задания', icon: 'assignment', url: '/tasks'},
        {title: 'Проекты', icon: 'work', url: '/projects'},
        {title: 'Личный кабинет', icon: 'account_circle', url: '/account'}
      ]
    }
    return [
      {title: 'Вход', icon: 'account_circle', url: '/login'},
      {title: 'Регистрация', icon: 'face', url: '/reg'}
    ]
  },
  methods: {
    closeError () {
      this.$store.dispatch('clearError')
    },
    onLogout () {
      this.$store.dispatch('logoutUser')
      this.$router.push('/')
    }
  }
}
</script>
```

Index.js

```
export default new Router({
  routes: [
    { path: '/',
      name: 'home',
```


Продолжение приложения Б

```
    component: Home},
  {  path: '/account',
    name: 'account',
    component: Account,
    beforeEnter: AuthGuard },
  {  path: '/login',
    name: 'login',
    component: Login},
  {  path: '/reg',
    name: 'reg',
    component: Reg },
  {  path: '/tasks',
    name: 'tasks',
    component: Tasks,
    beforeEnter: AuthGuard},
  {  path: '/projects',
    name: 'projects',
    component: Projects,
    beforeEnter: AuthGuard},
  {  path: '/projectTasks/:id',
    name: 'projectTasks',
    component: ProjectTasks,
    beforeEnter: AuthGuard},
  {  path: '/newProject',
    name: 'newProject',
    component: NewProject,
    beforeEnter: AuthGuard
  },
  {  path: '/newTask',
    name: 'newTask',
    component: NewTask,
    beforeEnter: AuthGuard},
  {  path: '/task/:id',
    props: true,
    name: 'task',
    component: Task}
],
mode: 'history'))
```

Продолжение приложения Б

Create_users_table.php

```
class CreateUsersTable extends Migration
{
    public function up()
    {
        Schema::create('users', function (Blueprint $table) {
            $table->id();
            $table->string('name');
            $table->string('surname')->nullable();
            $table->string('email')->unique();
            $table->timestamp('email_verified_at')->nullable();
            $table->string('password');
            $table->integer('position')->nullable();
            $table->rememberToken();
            $table->timestamps();
        });
    }
    public function down()
    {
        Schema::dropIfExists('users');
    }
}
```

Create_tasks_table.php

```
class CreateTasksTable extends Migration
{
    public function up()
    {
        Schema::create('tasks', function (Blueprint $table) {
            $table->id();
            $table->string('name');
            $table->text('description');
            $table->date('start');
            $table->date('end');
            $table->integer('executor');
            $table->integer('project');
            $table->timestamps();
        });
    }
    public function down()
    {
        Schema::dropIfExists('tasks');
    }
}
```

RouteServiceProvider.php

```
class RouteServiceProvider extends ServiceProvider
{
    @var string
    protected $namespace = 'App\Http\Controllers';
    @var string
```

Продолжение приложения Б

```
public const HOME = '/home';
    @return void
    public function boot()
    {    parent::boot();    }
    @return void
    public function map()
    {
        $this->mapApiRoutes();
        $this->mapWebRoutes();
    }
    @return void
protected function mapWebRoutes()
{
    Route::middleware('web')
        ->namespace($this->namespace)
        ->group(base_path('routes/web.php'));
}
    @return void
protected function mapApiRoutes()
{
    Route::prefix('api')
        ->middleware('api')
        ->namespace($this->namespace)
        ->group(base_path('routes/api.php'));    }}
}
```

Приложение В Акт внедрения

«ЭлитСтройПроект-КС»
ЖАУАПКЕРШІЛІГІ
ШЕКТЕУЛІ
СЕРІКТЕСТІГІ
(«ЭлитСтройПроект-КС»
ЖСШ)



ТОВАРИЩЕСТВО С
ОГРАНИЧЕННОЙ
ОТВЕТСТВЕННОСТЬЮ
«ЭлитСтройПроект-КС»
(ТОО «ЭлитСтройПроект-КС»)

Қазақстан Республикасы,
050046, Алматы қ.
Сатпаев к-сі, 90-үй, 904 -оф
E-mail: info@elitestroy.kz
Тел.: +7 (727) 311-99-22

Юридический адрес:
Республика Казахстан,
050046, Бостандыкский район, г. Алматы,
ул. Сатпаева, д. 90, оф. 904
E-mail: info@elitestroy.kz
Тел.: +7 (727) 311-99-22

Н.О.С. 2020 № 108

Акт внедрения информационной системы по управлению проектами «Elite»

Настоящий Акт свидетельствует о том, что информационная система по управлению проектами «Elite», разработанная Чертковой Натальей, внедрена в ТОО «ЭлитСтройПроект-КС». Система предназначена для автоматизации и упрощения процесса управления работой проектной группы, а также повышения производительности сотрудников за счет тайм-менеджмента и визуализации текущих задач.

Процесс внедрения проходил с 4 по 8 мая 2020г.

Заявленные характеристики системы предполагали наличие следующих основных возможностей:

- просмотр, создание, редактирование, удаление задач для выполнения;
- просмотр основной информации, по проектам, в которых пользователь принимает участие;
- прикрепление файлов по проектам и задачам;
- перемещение задач по стадиям;
- назначение задач конкретным пользователям.

В ходе опытной эксплуатации системы подтверждено, что она обладает всеми заявленными возможностями.

Показатели эффективности программы:

- оптимизация процесса управления работой проектной группы;
- сокращение издержек за счет автоматизации процесса управления;
- экономический эффект.

На момент подписания настоящего Акта система установлена и эксплуатируется сотрудниками данной компании. Программой пользуются 20 человек.

Директор ТОО «ЭлитСтройПроект-КС»



Ширяев В.А.

Исполнитель

Черткова Н.О.