**КЫРГЫЗ РЕСПУБЛИКАСЫНЫН БИЛИМ БЕРҮҮ ЖАНА ИЛИМ МИНИСТРЛИГИ**

**И.РАЗЗАКОВ атындагы КЫРГЫЗ МАМЛЕКЕТТИК ТЕХНИКАЛЫК УНИВЕРСИТЕТИ**

**ЭЛЕКТРОНИКА ЖАНА ТЕЛЕКОММУНИКАЦИЯЛАР ИНСТИТУТУ**

**«ТЕЛЕКОММУНИКАЦИЯДАГЫ МААЛЫМАТТАР СИСТЕМАЛАРЫ ЖАНА ТЕХНОЛОГИЯЛАРЫ» кафедрасы**

**«Маалыматтар системалары жана технологиялары» даярдоо багыты боюнча«Телекоммуникациядагы маалыматтар системалары жана технологиялары» профили боюнча**

**БҮТҮРҮҮЧҮ КВАЛИФИКАЦИЯЛЫК ИШ**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

(*Бүтүрүүчү квалификациялык иштин темасы*)

|  |  |  |
| --- | --- | --- |
| Аткарган | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  (кол белгиси) |
| Жумуштун жетекчиси | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  (кол белгиси) |
| “Жактоого уруксат берилди”  Кафедра башчысы | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  (кол белгиси) |
| Кафедра башчысы | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  (кол белгиси) |

Бишкек 2023

**МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ КЫРГЫЗСКОЙ РЕСПУБЛИКИ**

**КЫРГЫЗСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ им. И.РАЗЗАКОВА**

**ИНСТИТУТ ЭЛЕКТРОНИКИ И ТЕЛЕКОММУНИКАЦИЙ**

**Кафедра «ИНФОРМАЦИОННЫЕ СИСТЕМЫ И ТЕХНОЛОГИИ** **В ТЕЛЕКОММУНИКАЦИЯХ»**

**ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА**

**по направлению «Информационные системы и технологии»**

**профиль «Информационные системы и технологии в телекоммуникациях»**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

(*Тема выпускной квалификационной работы*)

|  |  |  |
| --- | --- | --- |
| Выполнил(а): | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  (подпись) |
| Руководитель ВКР | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  (подпись) |
| “Допущен к защите”  Заведующий кафедрой | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  (подпись) |
| Рецензент | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  (подпись) |

Бишкек 2023

И.РАЗЗАКОВ АТЫНДАГЫ КЫРГЫЗ МАМЛЕКЕТТИК ТЕХНИКАЛЫК УНИВЕРСИТЕТИ

КЫРГЫЗСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ

им. И.РАЗЗАКОВА

|  |
| --- |
| минбары (кафедрасы) |
| Кафедра |

|  |  |
| --- | --- |
|  | «БЕКИТЕМ»  «УТВЕРЖДАЮ»  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  Минбардын (кафедранын) башчысы  Зав.кафедрой \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  «\_\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ 20\_\_\_г. |

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ тобунун студенти

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(аты, ата-теги)

**ТАПШЫРМА**

**ЗАДАНИЕ**

на выпускную квалификационную работу студенту группы

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(фамилия, имя, отчетсво)

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

1. Квалификациялык жумуштун нуктасы:

Тема проекта:

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_ ж. \_\_\_\_\_\_\_\_\_ “\_\_\_\_\_” № факультеттин буйругу менен бекитилген.

утверждена приказом по факультету № \_\_\_\_\_\_\_\_\_ от “\_\_\_\_\_\_\_\_\_\_\_\_”

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_20\_\_г.

2. Студенттин жумушту тапшыруу мөөнөтү.

Срок сдачи студентом законченной работы \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

3. Квалификациялык жумушка карата алгачкы маалымат

Исходные данные к работе: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

|  |  |  |  |
| --- | --- | --- | --- |
| Катар №  № п/п | Эсептеп түшүндүрмө каттын мазмуну (иштетүүгө тийиштүү маселелердин тизмеси)  Содержание расчетно-пояснительной записки (перечень подлежащих разработке вопросов) | Көлөмү % менен  Объем в % | Аткаруу мөөнөтү  Срок  выполнения |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Катар №  № п/п | Чийим бөлүгү  Графическая часть | 24 көлөмдүү барагы  Объем в % | Көлөмү % менен  Объем в % | Аткаруу мөөнөтү  Срок  выполнения |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

**Айрым бөлүктөрү боюнча кеңеш берүүчүлөр (жетекчисинен тышкары)**

**Консультация по отдельным разделам (помимо руководителя)**

|  |  |  |
| --- | --- | --- |
| Катар №  № п/п | Бөлүктөрү (аталышы)  Раздел (наименование) | Кеңеш берүүчүлөрдүн аты, ата-теги жана кол белгиси  Ф.И.О. консультанта и подпись |
| 1. | Экономическая часть |  |
| 2. | Н.контроль |  |

Тапшырма берген күнү

Дата выдачи задания

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Жетекчи

Руководитель

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

|  |  |
| --- | --- |
|  | Аты, ата-теги, окумуштуу даражасы, наамы, кол белгиси  Ф.И.О., ученая степень, звание \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  (подпись) |

Эскертүү: Бул тапшырма бүткөн жумушка тиркелет жана мамлекеттик ымтыканга көрсөтүлөт.

Примечание: Это задание прилагается к законченной работе и предъявляется на ГАК.

Аннотация

Максаты: «Газпром Кыргызстан» компаниясынын калкка кызмат көрсөтүү процессин эффективдүү контролдоону жана башкарууну камсыз кылуу максатында тапшырмалардын аткарылышын эсепке алуу системасы иштелип чыккан. Ал тапшырмаларды түзүүгө жана көзөмөлдөөгө, сарпталган убакытты жана материалдарды эсепке алууга жана тандалган мөөнөттөр үчүн аткарылган тапшырмалардын статистикасын жалпылоого мүмкүндүк берет.

Өзгөчөлүк: Система компанияны кызматкерлердин жана подрядчылардын ишине ачык-айкындуулукту жана көзөмөлдү камсыз кылуучу эсепке алуу процессин автоматташтыруу үчүн инструменттерди берет. Бул уруксатсыз жабдууларды орнотуу жана мыйзамсыз киреше менен байланышкан мыйзамсыз иш-аракеттердин алдын алууга жардам берет.

Актуалдуулугу: Үйлөрдү жана батирлерди газдаштырууга муктаждыктын өсүп жаткан шартында процессти эффективдүү башкаруу жана көзөмөлдөө мындай ири компания үчүн маанилүү факторлор болуп саналат. Мыйзамсыз аракеттерге жол бербөө жана аткаруучулардын ишинин ачык-айкындуулугун камсыз кылуу зарылчылыгы барган сайын маанилүү болуп баратат. Тапшырмалардын аткарылышын көзөмөлдөө системасы компаниянын ишинин натыйжалуулугун жогорулатууга жана тапшырмаларды аткаруунун жогорку сапатын камсыз кылууга жардам берет.

Натыйжалуулугу: милдеттерди аткаруу эсепке алуу системасы компания төмөнкү натыйжаларга жетишүүгө мүмкүндүк берет:

* Объекттерди газдаштыруу процессинин айкындуулугун жана көзөмөлүн жогорулатуу.
* Мыйзамсыз иш-аракеттерди жана жабдууларды мыйзамсыз пайдаланууну алдын алуу.
* тапшырманы аткаруу убактысын жана материалды колдонууну оптималдаштыруу.
* Кызматкерлердин жана подряддык уюмдардын натыйжалуулугун жогорулатуу.
* Тапшырмаларды аткаруунун сапатын жана кардарлардын канааттануусун жогорулатуу.

Бул жыйынтыктардын баары Газпромдун натыйжалуу иштешине жана газдаштыруу боюнча кызмат көрсөтүүлөрдүн сапатын жогорулатууга алып келет.

Аннотация

Назначение: Система учета выполнения задач разработана для компании «Газпром Кыргызстан» с целью обеспечения эффективного контроля и управления процессом предоставления населению своих услуг. Она позволяет создавать и отслеживать задачи, учитывать затраченное время и материалы, а также подводить статистику выполненных задач за выбранные периоды.

Характеристика: Система предоставляет компании инструменты для автоматизации процесса учета выполнения задач, обеспечивая прозрачность и контроль над работой сотрудников и подрядчиков. Она помогает предотвратить нелегальные действия, связанные с установкой несанкционированного оборудования и незаконным получением прибыли.

Актуальность: В условиях растущей потребности в газификации домов и квартир, эффективное управление и контроль процессом являются важными факторами для такой крупной компании. Необходимость предотвращения нелегальных действий и обеспечения прозрачности работы исполнителей становится все более актуальной. Система учета выполнения задач поможет компании повысить эффективность своей деятельности и обеспечить высокое качество выполнения задач.

Эффективность: Система учета выполнения задач позволит компании достичь следующих результатов:

* Увеличение прозрачности и контроля над процессом газификации объектов.
* Предотвращение нелегальных действий и незаконного использования оборудования.
* Оптимизация времени выполнения задач и использования материалов.
* Улучшение эффективности работы сотрудников и подрядчиков.
* Повышение качества выполнения задач и удовлетворенности клиентов.

Все эти результаты приведут к более эффективному функционированию компании «Газпром» и повысят ее качество предоставления услуг газификации.

Оглавление

[Список используемых сокращений и обозначений 9](#_Toc135613161)

[Введение 10](#_Toc135613162)

[Глава 1. Аналитическая часть 11](#_Toc135613163)

[1.1 Обзор и анализ существующих аналогичных систем 11](#_Toc135613164)

[1.1.1 Atlassian Jira 11](#_Toc135613165)

[1.1.2 SAP (Systems, Applications, and Products) 13](#_Toc135613166)

[1.1.3 1С 15](#_Toc135613167)

[1.2 Обоснование необходимости и цели использования системы учета выполненных задач 17](#_Toc135613168)

[1.2.1 Декомпозиция задачи (IDEF0) 17](#_Toc135613169)

[1.2.2 Диаграмма потоков данных (DFD) 20](#_Toc135613170)

[1.2.3 Диаграмма вариантов использования (USE-CASE) 22](#_Toc135613171)

[1.3 Постановка задачи 24](#_Toc135613172)

[1.3.1 Цель и назначение создания «системы учета выполненных задач». Функциональные требования. 24](#_Toc135613173)

[1.3.2 Анализ требований к информационно-коммуникационным средствам для решения задачи. Нефункциональные требования. 27](#_Toc135613174)

[1.4 Анализ и обоснование выбора технологий 28](#_Toc135613175)

[Обоснование выбора технологий 28](#_Toc135613176)

[~~1.5 Обоснование среды разработки информационно-коммуникационной системы~~ 30](#_Toc135613177)

[Глава 2. Проектная часть 30](#_Toc135613178)

[2.1 Архитектура системы 30](#_Toc135613179)

[2.2 Пользовательский интерфейс 30](#_Toc135613180)

[2.3 Логическая часть 30](#_Toc135613181)

[2.4 Сторонние модули 30](#_Toc135613182)

[2.5 Интеграционная часть 30](#_Toc135613183)

# Список используемых сокращений и обозначений

* БД: База Данных
* (Р)СУБД: (Реляционная) Система управления базами данных
* ИС: Информационная система
* API (АПИ): Application Programming Interface - интерфейс прикладного программирования
* KPI: key performance indicators - ключевые показатели эффективности
* САПР: Система Автоматизированного Проектирования.
* ОС: Операционная Система.
* ИТ: Информационные Технологии.
* ПО: Программное Обеспечение.
* ООП: Объектно-Ориентированное Программирование.
* UI: User Interface - пользовательский интерфейс.
* UX: User Experience - пользовательский опыт.
* SQL: Structured Query Language - язык структурированных запросов.
* HTML: Hypertext Markup Language - язык гипертекстовой разметки.
* CSS: Cascading Style Sheets - каскадные таблицы стилей.
* JSON: JavaScript Object Notation - формат обмена данными, основанный на синтаксисе объектов JavaScript.
* REST: Representational State Transfer - архитектурный стиль для построения распределенных систем.
* URL: Uniform Resource Locator - унифицированный указатель ресурса (веб-адрес).
* IDE: Integrated Development Environment - интегрированная среда разработки.
* HTTP: Hypertext Transfer Protocol - протокол передачи гипертекста. HTTPS: Hypertext Transfer Protocol Secure - защищенная версия протокола HTTP.
* CRUD: Create, Read, Update, Delete - базовые операции над данными (создание, чтение, обновление, удаление).

# Введение

Введение служит, прежде всего, для определения места ВКР в ряду аналогичных разработок, для краткого, вводного ознакомления с сутью работы. В нем может быть дано краткое обоснование актуальности выбранной темы, сформулировано назначение разработки, приведены цели, которые планировалось достичь, и решаемые при этом задачи, указаны основные особенности реализованных решений, их новизна.

Рекомендуется писать введение по завершении основных глав работы, перед заключением. В этом случае исключена возможность несоответствия "желаемого" и "действительного".

TODO: ОПИСАТЬ К ЗАВЕРШЕНИЮ

Основная часть

# Глава 1. Аналитическая часть

## 1.1 Обзор и анализ существующих аналогичных систем

Существует множество систем, позволяющих осуществлять отслеживание и ведение текущих задач, многие из них обладают гибкостью и разнообразными функциональными возможностями, некоторые из них рассмотрены ниже.

### 1.1.1 Atlassian Jira

Это мощная и гибкая система управления задачами и проектами, которая предлагает широкий набор функциональных возможностей. Вот основные характеристики и функциональности Atlassian Jira:

1. Управление задачами и проектами: Jira предоставляет возможность создавать и отслеживать задачи, присваивать им исполнителей, устанавливать сроки выполнения и отслеживать прогресс. Можно организовывать задачи в виде проектов и структурировать их по различным категориям.
2. Отслеживание ошибок и запросов: Jira позволяет легко отслеживать и управлять ошибками, дефектами и запросами на улучшение. Можно создавать тикеты для проблем, назначать их разработчикам, отслеживать статус исправления и вести дискуссии внутри системы.
3. Гибкая система рабочих процессов: Jira позволяет настраивать и адаптировать рабочие процессы под требования вашей организации. Можно создавать собственные шаблоны задач, определять этапы выполнения и устанавливать правила перехода между ними.
4. Визуализация данных: Jira предоставляет графические диаграммы и дашборды, которые помогают визуализировать и анализировать данные о задачах и проектах. Можно создавать диаграммы Ганта, диаграммы burndown (оставшиеся задачи), сводные таблицы и другие типы отчетов для мониторинга прогресса и производительности.
5. Совместная работа и коммуникация: Jira предоставляет возможность совместной работы и коммуникации внутри команды. Можно комментировать задачи, упоминать коллег и обсуждать проблемы прямо в контексте задачи. Также, Jira интегрируется с другими инструментами для коммуникации, такими как Slack или Microsoft Teams.
6. Расширяемость и интеграция: Jira обладает широким набором плагинов и расширений, которые позволяют настроить систему под свои потребности. Она также интегрируется с другими популярными инструментами разработки, такими как Git, Confluence, Bitbucket и др.

Atlassian Jira является мощным инструментом для управления задачами и проектами, обладающим гибкостью и разнообразными функциональными возможностями. Он позволяет командам эффективно управлять рабочим процессом, отслеживать прогресс и улучшать коммуникацию внутри команды.

Несмотря на множество преимуществ, система Atlassian Jira также имеет некоторые недостатки, которые следует учитывать:

1. Сложность использования: Начальная настройка и освоение Jira могут быть сложными для новых пользователей. Интерфейс и функциональность системы могут показаться сложными для неподготовленных пользователей, особенно если они не имеют опыта работы с подобными инструментами.
2. Высокая стоимость: Jira является коммерческим продуктом, и его лицензирование может быть затратным, особенно для малых компаний или небольших проектов с ограниченным бюджетом. Дополнительные плагины или расширения также могут потребовать дополнительных затрат.
3. Необходимость настройки: Хотя Jira предлагает широкий набор функциональных возможностей, некоторые из них могут потребовать дополнительной настройки или использования плагинов. Это может повлечь за собой дополнительные затраты времени и ресурсов для настройки и интеграции системы.
4. Ограничения масштабируемости: В некоторых случаях, при работе с большим объемом данных или сложными рабочими процессами, Jira может столкнуться с ограничениями масштабируемости и производительности. Это может привести к задержкам или снижению производительности системы.
5. Необходимость обучения пользователей: Внедрение Jira в организации может потребовать дополнительного обучения пользователей. Это может быть вызовом, особенно если в организации есть сотрудники с ограниченными навыками работы с компьютером или недостаточным пониманием процессов управления задачами.

Итого, Atlassian Jira громоздкая система, с множеством возможностей, требует длительной подготовки перед ее настройкой и использованием, имеет проблемы с масштабированием и репликацией, а стоимость лицензий системы и ее плагинов в некоторых случаях, делают ее применение нецелесообразным с экономической точки зрения.

### 1.1.2 SAP (Systems, Applications, and Products)

SAP (Systems, Applications, and Products) — это крупная международная компания, предлагающая широкий спектр программных продуктов для управления бизнес-процессами в различных отраслях. Одним из ключевых продуктов SAP является платформа SAP ERP (Enterprise Resource Planning), которая предоставляет мощные функциональные возможности для ведения задач и управления бизнес-процессами. Вот некоторые из основных характеристик и функциональных возможностей SAP в плане ведения задач:

1. Интеграция процессов: SAP позволяет интегрировать различные бизнес-процессы, связанные с ведением задач, в единую систему. Это позволяет автоматизировать и упростить выполнение задач и обеспечивает единый источник данных для анализа и принятия решений.
2. Управление проектами: SAP ERP предоставляет функциональность для планирования, отслеживания и управления проектами. Это включает в себя возможности для определения задач, назначения ответственных лиц, установки сроков выполнения, контроля прогресса и оценки ресурсов.
3. Бюджетирование и учет затрат: SAP позволяет вести учет бюджетов проектов и задач, отслеживать затраты на материалы, трудовые ресурсы и другие расходы. Это помогает контролировать затраты, оптимизировать бюджеты и анализировать эффективность задач.
4. Коллаборация и коммуникация: SAP предоставляет инструменты для коллаборации и коммуникации внутри организации. Это включает в себя возможность обмена сообщениями, совместной работы над документами, управления комментариями и задачами, а также интеграцию с электронной почтой и другими коммуникационными средствами.
5. Аналитика и отчетность: SAP предлагает широкий набор инструментов для анализа данных и создания отчетов. Это позволяет анализировать производительность задач, оценивать качество выполнения работ, отслеживать ключевые показатели производительности и принимать информированные решения.
6. Персонализация и гибкость: SAP ERP позволяет настраивать и адаптировать систему под конкретные потребности организации. Это включает возможность настройки рабочего пространства, пользовательских ролей и доступа к функциональности, чтобы обеспечить оптимальное использование системы.

SAP предоставляет мощные инструменты для ведения задач и управления бизнес-процессами. Однако, конкретные характеристики и функциональные возможности могут варьироваться в зависимости от конкретной версии и конфигурации продукта SAP, а также от требований и настроек организации.

Недостатки SAP:

1. Сложность внедрения: Внедрение SAP может быть сложным и требовательным процессом. Оно может потребовать значительных усилий, времени и ресурсов, включая специалистов по SAP и консультантов. Это может повлечь высокие затраты на внедрение и обслуживание системы.
2. Высокая стоимость: SAP является мощной и расширяемой системой, однако она может быть дорогой для небольших и средних предприятий. Лицензирование и поддержка SAP могут иметь значительные затраты, что делает его недоступным для некоторых компаний с ограниченными бюджетами.
3. Сложность использования: Из-за масштабного функционала и гибкости SAP может быть сложным в использовании для новых пользователей. Необходимо обучение и опыт для эффективного использования системы. Это может потребовать времени и усилий для обучения сотрудников, и адаптации к новой среде работы.
4. Высокая зависимость от инфраструктуры: Для работы SAP требуется надежная и мощная IT-инфраструктура, включая серверы, базы данных и сетевые ресурсы. Это может потребовать дополнительных инвестиций в оборудование и поддержку IT-инфраструктуры.
5. Ограничения по настройке и гибкости: Некоторые пользователи могут столкнуться с ограничениями в настройке и гибкости SAP. Внесение изменений в систему или настройка специфических требований может быть сложным или требовать дополнительных ресурсов. Это может быть проблемой для организаций с уникальными бизнес-процессами или специфическими требованиями.

Итого, SAP еще более сложная в освоении, чем Atlassian Jira, более требовательная к инфраструктуре, также платная, имеет меньшую распространенность, за счет чего найти квалифицированного специалиста для ее поддержки (за адекватную цену) почти невозможно.

### 1.1.3 1С

Это платформа и набор программных продуктов, предназначенных для автоматизации бизнес-процессов и управления предприятием. В контексте ведения задач, 1С предлагает следующие основные характеристики и функциональные возможности:

1. Управление задачами: 1С позволяет создавать и отслеживать задачи, определять их приоритеты, статусы, сроки выполнения и ответственных исполнителей. Можно назначать задачи конкретным сотрудникам и отслеживать их выполнение.
2. Календарное планирование: 1С предоставляет возможность создания календарей и планирования задач на определенные даты и сроки. Можно определить зависимости между задачами, устанавливать предшествующие и последующие задачи, а также распределить ресурсы и определить сроки выполнения.
3. Управление ресурсами: 1С позволяет управлять ресурсами, связанными с выполнением задач. Можно назначать сотрудников, материальные и финансовые ресурсы для каждой задачи. Это позволяет контролировать расходы и использование ресурсов в процессе выполнения задач.
4. Коммуникация: 1С предоставляет инструменты для коммуникации и совместной работы над задачами. Можно обмениваться сообщениями, комментариями, прикреплять файлы и делиться информацией с другими участниками проекта. Это способствует эффективной командной работе и обеспечивает прозрачность процесса выполнения задач.
5. Отчетность и аналитика: 1С предоставляет возможность получать отчеты и аналитическую информацию о выполнении задач. Можно анализировать статусы, сроки, затраты и другие параметры задач для принятия решений и оптимизации процесса выполнения.
6. Интеграция с другими системами: 1С обладает возможностью интеграции с другими системами и программными продуктами. Можно интегрировать 1С с почтовыми сервисами, электронными документами, системами управления проектами и другими решениями для обеспечения более полного и связанного рабочего процесса.

Недостатки 1С:

1. Сложность настройки и разработки: При работе с 1С может потребоваться определенный уровень знаний и опыта для настройки и разработки функциональности. Создание и настройка сложных бизнес-процессов может требовать углубленного понимания конфигураций и языка программирования 1С.
2. Высокая стоимость: Лицензирование и внедрение 1С могут быть затратными. Стоимость лицензий, обновлений и поддержки может быть значительной, особенно для больших предприятий или специализированных конфигураций.
3. Ограниченные возможности интеграции: Встроенные инструменты интеграции в 1С могут быть ограничены. Это может затруднять интеграцию с другими системами, особенно с технологически сложными или специализированными решениями.
4. Зависимость от поставщика: 1С является проприетарной системой, что означает зависимость от компании-разработчика. Изменение поставщика или переход на другую платформу может быть сложным и затратным процессом.
5. Ограниченная масштабируемость: Некоторые версии 1С могут иметь ограничения по масштабируемости, особенно при работе с большим объемом данных или высокой нагрузкой. Это может потребовать дополнительных усилий для оптимизации производительности и масштабирования системы.
6. Сложности при обновлении: При переходе на новые версии 1С могут возникать сложности и проблемы совместимости с предыдущими настройками и разработками. Обновление системы может потребовать тщательного планирования и тестирования.
7. Ограниченная гибкость: Некоторые пользователи могут считать, что 1С имеет ограниченные возможности для настройки под свои индивидуальные потребности и процессы. Это может быть особенно заметно в случае использования стандартных конфигураций, которые не всегда могут удовлетворять специфическим требованиям организации.

1С имеет множество возможностей, которые могут различаться в зависимости от конкретной конфигурации и модулей. Но, самый большой недостаток при выборе 1С - это негативный опыт ее использования компанией «Газпром Кыргызстан».

Есть еще множество систем, способных в той или иной степени осуществлять - учет и ведение задач компании «Газпром Кыргызстан», такие как: Redmine, GitLab, Bugzilla, Trac, которые не будут рассмотрены в данном документе.

чем будет отличаться проектируемая система или технология решения задачи от существующей

почему необходимо разрабатывать новое решение

перечислить основные факторы выбора, обосновать выбор применяемой технологии и дать особенности ее использования в данном проекте.

## 1.2 Обоснование необходимости и цели использования системы учета выполненных задач

### 1.2.1 Декомпозиция задачи (IDEF0)

Декомпозиция решения задачи позволяет разбить сложную задачу на более простые и управляемые компоненты, что облегчает понимание и реализацию системы.  
В рамках проведения декомпозиции, мы определяем основные функции и подзадачи, которые необходимо выполнить для достижения целей системы. Затем мы структурируем эти функции в иерархическую структуру, где каждый компонент системы отвечает за определенную функциональность.  
Для визуализации этой структуры и взаимосвязей между компонентами, мы используем диаграмму IDEF0. На данной диаграмме мы будем представлять функциональные блоки и связи между ними, а также входы и выходы каждого блока.  
Декомпозиция и создание диаграммы IDEF0 позволят нам лучше понять структуру системы, выделить ключевые компоненты и определить их взаимосвязи. Это будет полезной основой для дальнейшего проектирования и разработки информационной системы, а также для более эффективного управления задачами и ресурсами.

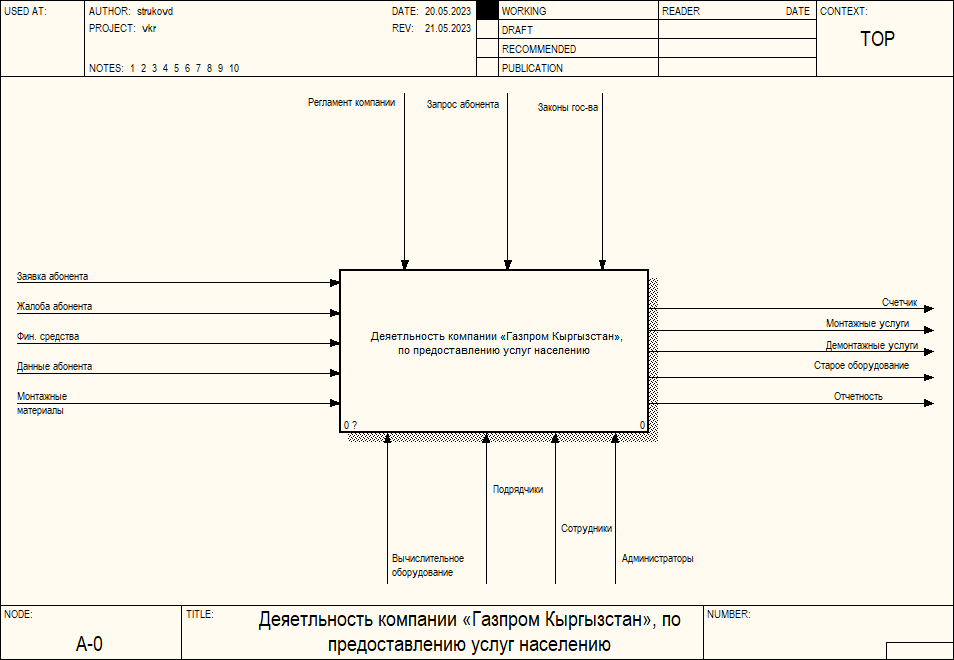


Рис. 1.1 Общая схема работы предприятия, по части обслуживания населения

На диаграмме видны основные входящие параметры, входящие из вне системы, снизу видны механизмы, позволяющие управлять процессом (администрация, оборудование, подрядчики, и прочие сотрудники), необходимые для работы, сверху видны правила, в соответствии с которыми выполняется работа (регламент, закон, и запрос абонента), и виден результат работы (новые абонентские счетчики, демонтажные услуги, монтажные услуги, отработанное оборудование после демонтажа, отчетность). Декомпозируем процесс, для рассмотрения деталей:

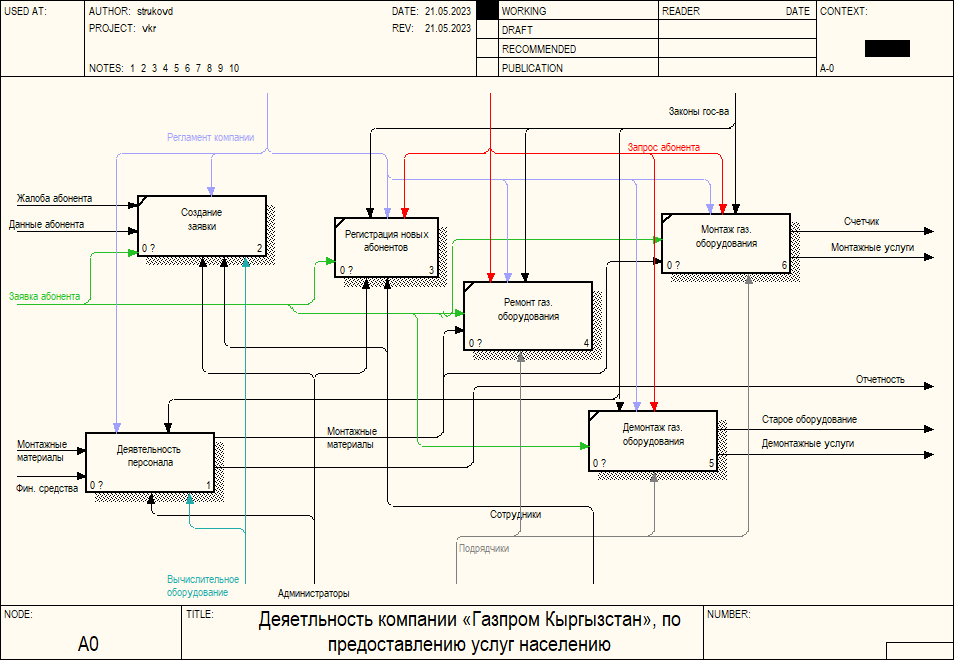


Рис. 1.2 Декомпозиция схемы работы предприятия, по части обслуживания населения.

На схеме видны основные функции, выполняющиеся в процессе работы, в целом это работает так: - создается заявка, по которой сотрудники и подрядчики, учитывая заявку абонента выполняют некие работы. Отдельно выделена работа администрации, ее декомпозиция рассмотрена ниже:

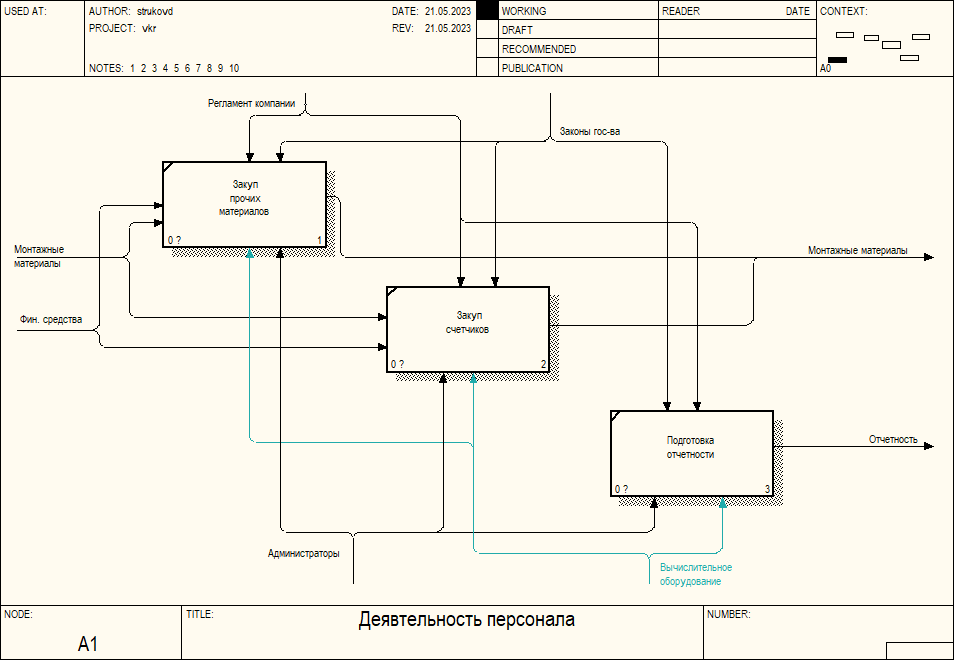


Рис. 1.3 Декомпозиция схемы работы предприятия, по части обслуживания населения.

В целом, на схеме отображено то, что администрация снабжает сотрудников необходимыми материалами, и следит за статистикой выполнения работ.

По приведенным диаграммам можно выявить основные функциональные требования (т.е. то, какие функции от системы требуются), а именно:

* Учет заявок
* Ведение учета абонентов (или связи с системой абонентов)
* Учет используемых материалов и оборудования
* Отчетность о проделанной работе (за определенный период)

Для более рассмотрения более приближенной к требованиям информационной системы и анализа потоков данных, приведем соответствующую диаграмму:

### 1.2.2 Диаграмма потоков данных (DFD)

В то время как **IDEF0 уделяет больше внимания функциональности** и взаимодействию компонентов системы, **DFD позволяет сфокусироваться на потоках данных и обработке информации** в информационной системе, она помогает понять, какие данные используются в системе, как они передаются между компонентами и как происходит их обработка. Важной особенностью DFD является то, что она не учитывает внутреннюю логику или подробности реализации каждого компонента, а сконцентрирована на потоках данных между ними. DFD позволяет выделить основные процессы, внешние сущности и данные, а также их взаимодействие.

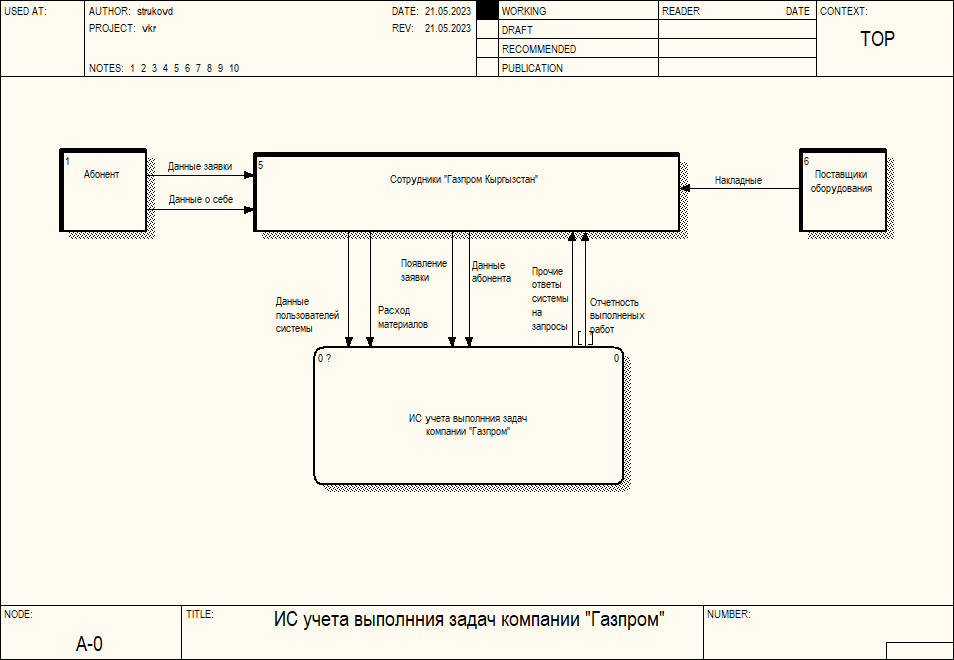


Рис. 2.1 Внешние потоки данных

На рисунке видно примеры входящих потоков данных входящий в систему и сущности, порождающие эти потоки. Ниже представлена декомпозиция, отображающая функции, которые используют входящие потоки данных.

Отчетность о проделанной работе (за определенный период)

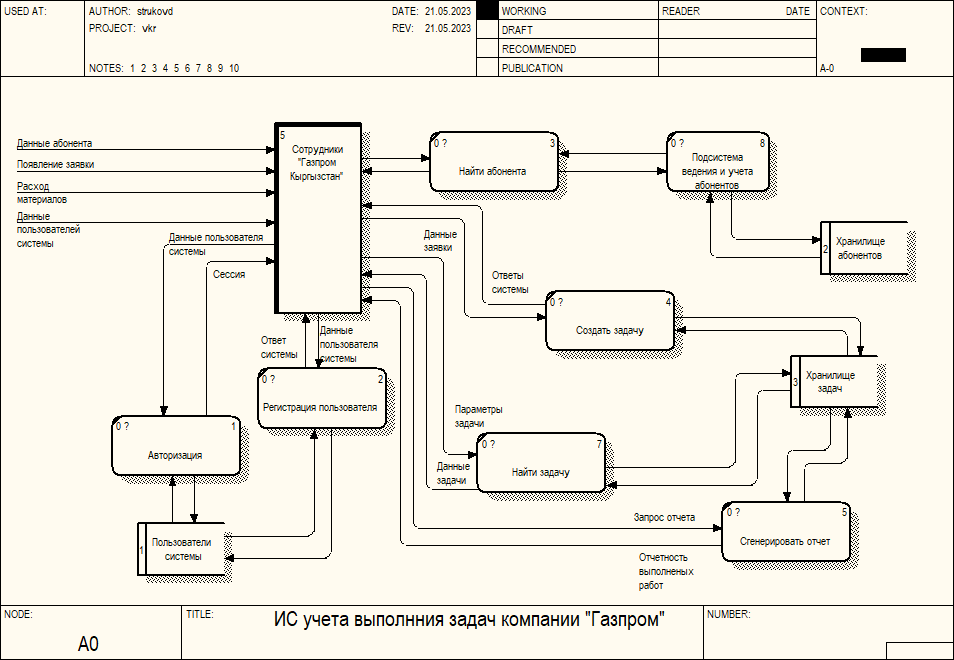


Рис. 2.2 Декомпозиция диаграммы потоков данных

По приведенным диаграммам можно выявить основные данные которые необходимо учитывать в системе, а именно:

* Данные абонентов (через подсистему)
* Данные пользователей
* Данные задач (заявок)

В итоге, мы можем сформулировать функциональные требования в виде специальной диаграммы вариантов использования.

### 1.2.3 Диаграмма вариантов использования (USE-CASE)

Диаграмма USE-CASE является эффективным инструментом для идентификации и моделирования требований к информационной системе. Она позволяет визуально представить различные сценарии использования системы и взаимодействие ее актеров с функциональными возможностями системы.

В представленной ниже диаграмме мы сосредоточимся на основных актерах системы и их ролях во взаимодействии с системой. Актеры могут быть пользователями, внешними системами, а также другими компонентами, которые взаимодействуют с нашей информационной системой.

Каждый вариант использования (USE-CASE) представляет собой конкретную функциональную возможность системы и сценарий, в котором актеры взаимодействуют с системой для достижения определенных целей. Мы опишем основные актеры и варианты использования, а также их связи и зависимости друг от друга.

Ранее, при рассмотрении диаграммы IDEF0, был представлен список основных функциональных требований (список функций, которые требуются от системы), диаграмма вариантов использования позволяет графически (более наглядно) представить наш список функциональных требований к системе и что позволит проанализировать и выявить основные сценарии использования системы и в дальнейшем спроектировать информационную систему, основываясь на ее данных

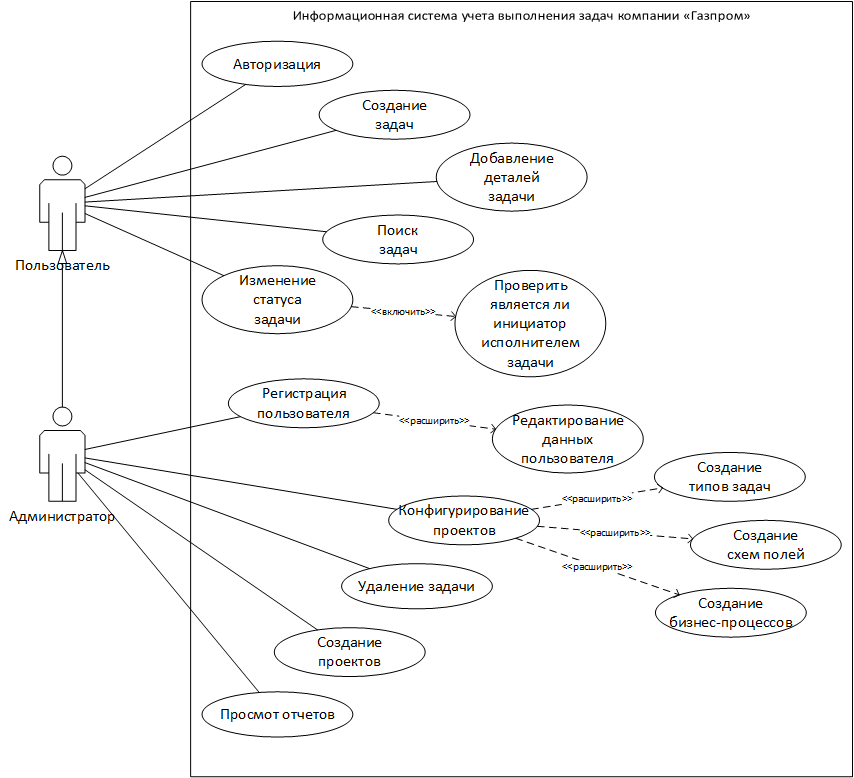


Рис 3.1 Диаграмма вариантов использования (USE-CASE)

На диаграмме видны основные функции, необходимые для работы системы, а также видны роли пользователей системы.

- выявить **основные** **недостатки**, **присущие существующей системе обмена** и обработки информации

## 1.3 Постановка задачи

сформулировать цель и задачи разработки и выделить основные требования к проектируемой системе обработки данных

- стоит определить тип проектируемой системы: это может быть телекоммуникационная система, **информационная система** централизованного хранения информации

### 1.3.1 Цель и назначение создания «системы учета выполненных задач». Функциональные требования.

Целью разработки является создание системы учета задач (упрощенный аналог Atlassian Jira), которая будет обеспечивать эффективное управление процессом выполнения задач, контроль времени выполнения, использованных материалов и генерацию статистики по срокам выполнения задач за выбранные периоды. Главной целью является повышение эффективности работы сотрудников и подрядчиков, предотвращение нелегальных практик и обеспечение надежной отчетности перед компанией «Газпром Кыргызстан».

#### Основные задачи разработки:

1. Разработать систему, позволяющую создавать и управлять задачами, включая возможность назначения ответственных лиц, приоритетов, дедлайнов и статусов задач.
2. Реализовать функционал учета времени выполнения задач, позволяющий отслеживать затраченное время на каждую задачу и составлять соответствующую статистику.
3. Обеспечить возможность учета использованных материалов при выполнении задач, таких как оборудование, инструменты и другие ресурсы.
4. Разработать механизм генерации статистических отчетов о сроках выполнения задач за выбранные периоды, позволяющий анализировать эффективность работы исполнителей и выявлять проблемные области.
5. Обеспечить безопасность и защиту данных системы, включая механизмы аутентификации, авторизации и защиты от несанкционированного доступа.
6. Реализовать удобный пользовательский интерфейс, обеспечивающий удобство использования системы и интуитивно понятную навигацию.

#### Функциональные требования:

Ранее мы уже выявляли функциональные требования, общие требования были выявлены после формирования диаграммы IDEF0, а далее более подробно представили их в виде диаграммы USE-CASE.

После выявления цели (назначения) системы, перечислим список необходимых функций:

* Создание задач: система должна позволять пользователям создавать новые задачи, указывая информацию о них, такую как заголовок, описание, сроки выполнения, ответственные лица и приоритет.
* Управление задачами: система должна предоставлять возможность управлять задачами, включая назначение ответственных, изменение статусов, установку приоритетов, отслеживание выполнения и уведомления о сроках.
* Комментирование задач: пользователи должны иметь возможность комментировать задачи, обмениваться информацией и обсуждать детали их выполнения.
* Отчетность: система должна предоставлять возможность генерировать отчеты о выполнении задач, статусе проектов, использовании ресурсов и других важных метриках.
* Уведомления и оповещения: система должна отправлять уведомления и оповещения пользователям о новых задачах, изменениях статусов, сроках выполнения и других важных событиях.
* Фильтрация и поиск: пользователи должны иметь возможность фильтровать и искать задачи по различным критериям, таким как статус, приоритет, ответственный и другие параметры.
* Интеграция с другими системами: система должна обеспечивать возможность интеграции с другими информационными системами, такими как системы учета времени, системы управления проектами и другими, для обмена данными и совместной работы.

#### Учет недостатков аналогичных систем:

Данная система позволит избежать недостатки приведенных аналогов в предыдущем разделе, а именно:

* Расширяемость: Собственная система разработки задач может быть разработана с учетом **будущих потребностей** и возможности расширения функциональности в дальнейшем. Можно добавлять новые модули, интегрировать с другими системами и адаптироваться к изменяющимся требованиям без зависимости от вендора или ограничений стороннего ПО.
* Узкоспециализированность: Разработка собственной системы позволяет более **точно соответствовать бизнес-требованиям**, анализировать и внедрять бизнес-процессы компании. Можно создать систему, полностью соответствующую уникальным требованиям, что повышает эффективность и улучшает рабочие процессы, и не требует **длительной подготовки персонала** перед ее использованием.
* Независимость от внешних поставщиков: Собственная разработка освобождает от зависимости от вендоров и сторонних поставщиков ПО. Мы имеем контроль процесса разработки, обслуживания и поддержки системы, а также имеете полный доступ к исходному коду и **интеллектуальной собственности**.
* Более низкие затраты в долгосрочной перспективе: Хотя начальные затраты на разработку собственной системы могут быть выше, в долгосрочной перспективе это может привести к снижению общих затрат. Собственная система не требует огромных вложений на **оплату лицензий** и не зависит от регулярных обновлений или дополнительных модулей, что может сэкономить средства на протяжении всего срока эксплуатации.

В этом разделе описывается то, какие функции должно предоставлять ПО (что система делает, и какие задачи выполняет)

Функции UI, операции базы данных, обработку данных, взаимодействие с другими системами и так далее. Функциональные требования обычно формулируются в виде конкретных задач, которые ПО должно успешно выполнить.

Цель решения задачи должна **сводиться к устранению тех недостатков, которые были отмечены автором в предыдущем разделе**, поэтому ее можно разделить на две группы подцелей:

- достижения улучшения ряда показателей выполнения выбранной производственной или управленческой функции или работы рассматриваемого подразделения, или всего предприятия в целом;

- улучшения значений показателей качества хранения, передачи и обработки информации.

При описании **назначения**, сделать акцент на перечень тех функций управления, которые **будут автоматизированы** при внедрении результатов.

В этом же подразделе формируются **функциональные требования к системе**. Они описывают **поведение системы и сервисы (функции)**, которые она должна выполнять.

### 1.3.2 Анализ требований к информационно-коммуникационным средствам для решения задачи. Нефункциональные требования.

Важно учесть не только функциональные требования, которые определяют функционал и возможности системы, но и нефункциональные требования, которые определяют качество и характеристики системы. Нефункциональные требования играют ключевую роль в обеспечении надежности, безопасности, производительности и удобства использования системы. Ниже перечислены основные нефункциональные требования, которые следует учесть при проектировании и разработке нашей информационной системы учета задач. Эти требованиями позволят нам определить приоритеты и сделать обоснованный выбор технологий и инструментов разработки.

1. Производительность: система должна обеспечивать высокую производительность и отзывчивость даже при большой нагрузке и объеме данных.
2. Надежность: система должна быть стабильной и надежной, предотвращать потерю данных и обеспечивать непрерывную работу.
3. Безопасность: система должна обеспечивать защиту данных и предотвращать несанкционированный доступ.
4. Масштабируемость: система должна быть способной масштабироваться и адаптироваться к росту объема данных и числа пользователей.
5. Удобство использования: система должна быть интуитивно понятной и легкой в использовании для пользователей разного уровня навыков.
6. Совместимость: система должна быть совместима с другими существующими системами и стандартами.
7. Поддержка и обслуживание: система должна быть легко поддерживаемой и обслуживаемой, с возможностью резервного копирования данных и мониторинга производительности.

В этом разделе рассматриваются требуемые аспекты качественные атрибуты и характеристики ПО. Такие как: производительность, надежность, уровень безопасности, удобство использования, масштабируемость и другие аспекты, которые влияют на качество и эффективность ПО.

Нефункциональные требования - характеристики системы и ее внешнего окружения.

Дополнительно могут **перечисляться ограничения**, накладываемые на действия и функции системы, а также на условия разработки (ограничения по времени, ограничения на организацию проекта, стандарты и т.д.)

## 1.4 Анализ и обоснование выбора технологий

Проведя анализ (нефункциональных) требований и степень моих навыков, было принято решение выбрать следующий стек технологий для написания модулей (сервисов):

* Node.js - основная язык программирования.
* Nest.js - удобный фреймворк для бэкенда.
* Vue.js - простой фреймворк для фронтенда.
* JSON + REST API - распространенный формат передачи данных.
* Elasticsearch - Оптимизированная СУБД для поиска больших объемов данных.
* Redis - быстрая СУБД, для хранения сессий.
* PostgreSQL - бесплатная и стабильная СУБД, для хранения данных задач и другой информации.
* Docker - контейниризация, для упрощения развертывания и безопасности.
* Linux - некоммерческая операционная система (CentOS).
* Микросервисная архитектура - для упрощения поддержки и работоспособности системы.

### Обоснование выбора технологий

#### Node.js

Является высокопроизводительной платформой для разработки серверной части. Он основан на событийно-ориентированной и неблокирующей архитектуре, что позволяет эффективно обрабатывать большое количество запросов и обеспечивать отзывчивость системы, что соответствует критерию нефункционально требования - **производительность**.

#### Микросервисная архитектура

Способствует высокой надежности и отказоустойчивости системы. Каждый модуль (сервис) может функционировать независимо, что минимизирует влияние сбоев и позволяет легко масштабировать и обслуживать отдельные компоненты. Микросервисная архитектура в сочетании с использованием Docker и Elasticsearch позволяет гибко **масштабировать** отдельные компоненты системы в зависимости от нагрузки. Мы можем горизонтально масштабировать сервисы, увеличивая количество экземпляров в контейнерах, а Elasticsearch позволяет обрабатывать большие объемы данных и запросов. Так же, в дальнейшем совместно с docker можно использовать оркестратор Kubernetes, для управления контейнеризированными сервисами, что также упростит масштабирование системы.

#### Nestjs

Позволяет удобно делить код на отдельные модули, включает множество архитектурных решений, и структурированный жизненный цикл запросов, включающий в себя шаблонные (паттерны) подходы такие как: guard, pipes, interceptors, декораторы, DI-систему, и так далее. Это позволяет удобно структурировать код, и минимизировать вероятность ошибок.

#### Docker

Обеспечивает изоляцию и контроль доступа к каждому контейнеру. Это повышает **безопасность** системы, так как в случае несанкционированного получения доступа к системе (взлома) сервисного контейнера, злоумышленник будет ограничен в своих возможностях и не сможет получить полный доступ к данным и ресурсам системы.

#### Vue.js

Предоставляет удобный и гибкий фреймворк для разработки интерфейса пользователя. Он предоставляет компонентный компонентный подход, что может позволить реализовать **удобный интерфейс**. Также, популярность Vue.js, простота в изучении и использовании позволит без особого труда найти специалиста для **поддержки** или доработки существующей системы в бедующем.

#### REST API

REST API является стандартным протоколом взаимодействия между клиентской и серверной частями системы. Это обеспечивает **совместимость с другими системами** и возможность интеграции с внешними сервисами.

Весь выбранный стек технологий имеет широкую поддержку сообщества разработчиков и предоставляет богатый выбор инструментов и библиотек для упрощения разработки, отладки и **обслуживания системы**.

redis, elastic, email, nestjs, nodejs,

схема микросервисов??

необходимо описание и обоснование выбора **технологии**, и **инструментарий (CASE средства)** проектирования будущей информационной системы.

## ~~1.5 Обоснование среды разработки информационно-коммуникационной системы~~

nestjs - модульность, docker - изоляция (от бэкдоров), linux (centos) - потому-что виртуализация, поддержка докера, .., git, сборка - typescript,

требуется представить обоснование выбора **среды и средств разработки** информационной системы. Описание **языка программирования**, анализ **достоинств** использования данной среды **для** разработки именно **данной системы**, описание **системы управления базы данных**.

git, nestjs, vuejs

# Глава 2. Проектная часть

## 2.1 Архитектура системы

## 2.2 Пользовательский интерфейс

## 2.3 Логическая часть

## 2.4 Сторонние модули

## 2.5 Интеграционная часть