МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ

федеральное государственное бюджетное образовательное учреждение

высшего образования «Казанский национальный исследовательский

технический университет им. А. Н. Туполева-КАИ»

(КНИТУ-КАИ)

Институт компьютерных технологий и защиты информации

Кафедра Прикладной математики и информатики

(01.03.02) Прикладная математика и информатика

Пояснительная записка к курсовой работе

по дисциплине «Проектирование и архитектура программных систем»

Тема: «Система контроля и учета выполнения задач и поручений»

Выполнили: студенты группы 4317  
Мохамед М.О

Проверил: доц. каф. ПМИ

Александров А. Ю.

Казань, 2023

**Оглавление**

[1. Введение 3](#_Toc105093124)

[2. Задание на курсовую работу 3](#_Toc105093125)

[3. Цели и задачи проекта 4](#_Toc105093126)

[4. Требования к программной системе 5](#_Toc105093127)

[4.1. Требования пользователя 5](#_Toc105093128)

[4.2. Функциональные требования 5](#_Toc105093129)

[4.3. Нефункциональные требования 6](#_Toc105093130)

[4.4. Требования к информационной безопасности 6](#_Toc105093131)

[5. Моделирование предметной области 6](#_Toc105093132)

[5.1. Выявление акторов системы и вариантов использования 7](#_Toc105093133)

[5.2. Диаграммы потоков данных 9](#_Toc105093134)

[5.3. Диаграммы сущность-связь 11](#_Toc105093135)

[5.4. SADT Диаграммы 12](#_Toc105093136)

[5.5. Диаграмма классов предметной области 15](#_Toc105093137)

[6. Проектирование общей архитектуры системы 17](#_Toc105093138)

[7. Описание программы 19](#_Toc105093139)

[7.1. Общие сведения 19](#_Toc105093140)

[7.2. Функциональное назначение 19](#_Toc105093141)

[8. Руководство пользователя 22](#_Toc105093144)

[8.1. Назначение программы 22](#_Toc105093145)

[8.2. Выполнение программы 25](#_Toc105093146)

[Заключение 27](#_Toc105093147)

[Глоссарий 28](#_Toc105093148)

[Приложение 1 **Ошибка! Закладка не определена.**](#_Toc105093149)

[Приложение 2 **Ошибка! Закладка не определена.**](#_Toc105093150)

[Приложение 3 **Ошибка! Закладка не определена.**](#_Toc105093151)

[Приложение 4 **Ошибка! Закладка не определена.**](#_Toc105093152)

# Введение

Современный бизнес невозможен без различного рода задач и назначения их соответствующему исполнителю. Их передача и назначение соответствующему исполнителю, своевременное реагирование на текущую обстановку, возможность получить обратную связь и на основании полученных данных скорректировать план действий для компании очень важно в современной жизни. Это позволяет комплексно увидеть ситуацию в компании, ее слабые и сильные стороны, что способствует росту и развитию компании.

С развитием IT-технологий появилась возможность упростить и даже автоматизировать передачу поручений и задач во избежание пустой траты времени. Ведь часто происходит так, что, ответственных за выполнение тех или иных заданий становится невозможно найти.

Именно для повышения эффективности выполнения выработанных задач необходима оптимизация их передачи.

Если учесть, что мы живём в век стремительного развития интернет-технологий, когда смартфоны и web-сервисы имеют наиболее широкое распространение, мы решили реализовать данную систему посредством фреймворка для создания мобильных приложений под Android и iOS, веб-приложений - Flutter.

# Задание на курсовую работу

Разработать систему контроля и учета выполнения задач и поручений проекта любой специализации.

# Цели и задачи проекта

Целями проекта является разработка и программная реализация контроля и учета выполнения задач и поручений проекта любой специализации, а также закрпление знаний, полученных при изучении дисциплины и получение практических навыков проектирования информационных систем с использованием современных технологий и инструментальных средств, и оформления сопровождающей документации программного обеспечения.

Среди задач проекта можно выделить следующие:

* создание базы данных, включающей в себя информацию о пользователях, их ролях, задачах внесенных в систему;
* разработка и создание android и ios приложения для взаимодействия системы с пользователями

Актуальность работы заключается в том, что создание системы контроля и учета выполнения задач и поручений в виде мобильного приложения и базы данных позволит в некоторой мере автоматизировать взаимодействие исполнителя и руководстводящего состава, что позволит получить больше прибыли и добиться большего успеха на общем рынке. Объектом нашей курсовой работы является создание системы контроля и учета выполнения задач и поручений.

Предметом курсовой работы являются Фреймворк Flutter для разработки мультиплатформенных приложений.

# Требования к программной системе

## Требования пользователя

* Поиск выполненных задач и ответственных за них сотрудников

Обеспечение Директору полной информации о выполненных и текущих задачах, их исполнителях, сроках, дополнительной информации

* Учет приоритета выполнения задачи

Внедрение оценки приоритета выполнения задачи, для повышения оперативности выполнения работы

* Автоматическое уведомление пользователей о сроках выполнения задач

Исключение лишней коммуникации сотрудников, обеспечение своевременного выполнения поставленных задач

* Возможность исполнителю выбора задачи для выполнения из списка «свободных» задач

Повышение эффективности труда исполнителя за счет возможности выбора наиболее подходящей для него задачи. Принцип открытого рынка

* Назначение задачи определенному пользователю

Исключение появления «висящих» задач

## Функциональные требования

* Регистрация пользователей, в том числе администратора, управляющего, пользователя

Выделение трех ролей в системе

* Создание задачи

Добавление задачи в базу данных

* Задание крайнего срока сдачи задачи

Добавление ключевых сведений о задаче в базу данных

* Уведомление исполнителей и менеджерах о сроках выполнения задач
* Учет списка задач
* Учет приоритета выполнения задачи
* Сдача отчета о выполнении задачи исполнителем менеджеру
* Проверка на срок выполнения задачи менеджером (вовремя, с опозданием)
* Хранение и использование информации о выполненных задачах
* Удаление задачи после проверки менеджером, перевод ее в «архив» выполненных задачах
* Доступ администратору ко всей информации о текущих и выполненных задачах

## Нефункциональные требования

* Структурированная информация
* Грамотный пользовательский интерфейс
* Приятный дизайн
* Легкость и простота использования
* Нормальная скорость открытия ресурса (не более 3 секунд)

## Требования к информационной безопасности

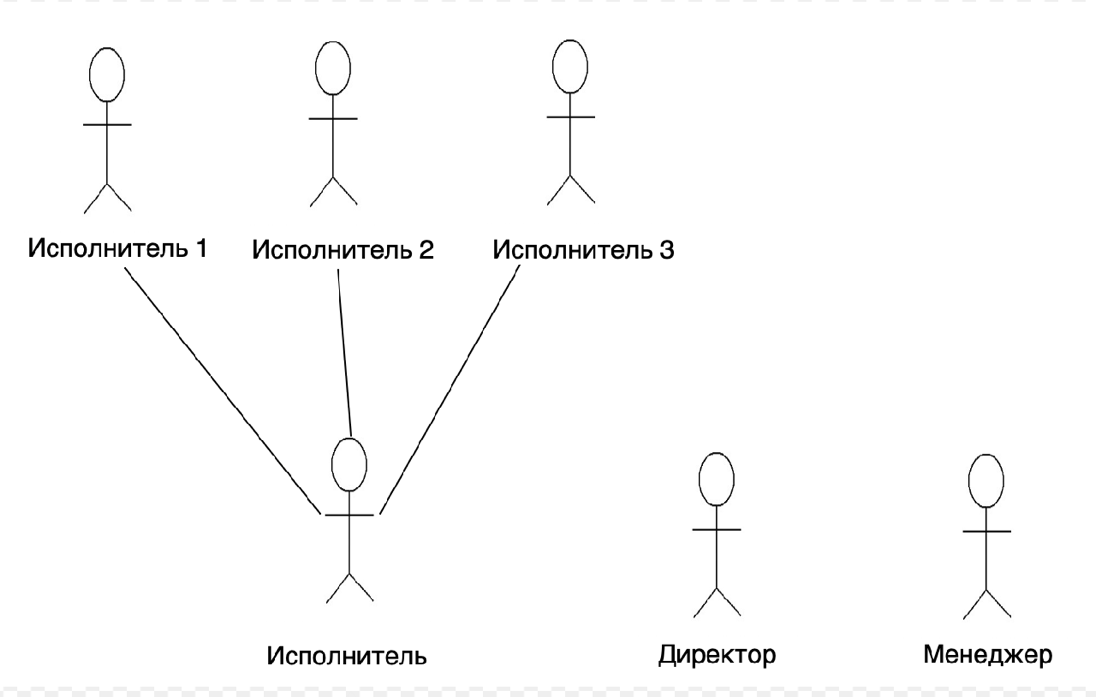
Отсутствуют

# Моделирование предметной области

Выявление акторов

1. **Выявление акторов**

На рисунке 1 представлены основные кандидаты в акторы системы.



Краткое описание акторов представлено в табл. 1.

**Табл. 1.** **Выявление акторов**

|  |  |
| --- | --- |
| *Актор* | *Краткое описание* |
| Директор | Получает задачи от заказчика в составе проекта или отдельно, в ходе его исполнения. Следит за их исполнением проекта. При необходимости может вносить оперативные корректировки в описания заказов, снять задачу с производства. |
| Менеджер | Разделяет проект на задачи, формирует план задачи (формирует план). Передаёт задачи к исполнению. Назначает исполнителей каждой из задач. Следит за фактическим исполнением работ. Анализирует фактическое исполнение планов. Вносит оперативные корректировки в планы. |
| Исполнитель | Выполняет назначенные задачи, отчитывается перед менеджером. |

# Выявление вариантов использования

Выявленные варианты использования сведены в таблицу 2.

**Табл. 2. Выявление вариантов использования**

|  |  |  |
| --- | --- | --- |
| *Основной актор* | *Наименование* | *Формулировка* |
| Директор | Внесение корректировок | Директор может откорректировать информацию новой или существующей задаче и назначенной на эту задачу исполнителе и передавать ее менеджерам |
| Директор | Запрос о задаче | Используется директором для поиска нужной информации о состоянии задачи в исполнении, необходимой для клиента. |
| Директор | Получение отчета | Используется директором для получения отчета об исполнении задачи, необходимой для клиента. |
| Директор | Изменение статуса исполнителя | Директор может повысить или понизить статус исполнителя, что влияет на выбор менеджера при назначении задачи исполнителю |
| Менеджер | Изменение задачи | Менеджер может откорректировать информацию о задаче в исполнении |
| Менеджер | Удаление задачи. | При необходимости снятия задачи с исполнения менеджер вызывает функцию «Удаление задачи». |
| Менеджер | Запрос о задаче | Используется менеджером для поиска нужной информации о состоянии задачи в исполнении, необходимой для клиента. |
| Менеджер | Планирование новой задачи | Менеджер размещает задачу в план в «хвост» очереди |
| Менеджер | Коррекция плана | Менеджер корректирует план при появлении каких-либо нестыковок |
| Менеджер | Планирование срочной задачи | Менеджер размещает задачу в требуемое время; очередь задач смещается |
| Менеджер | Выдача сменной задачи | Менеджер формирует сменную задачу для исполнителя |
| Менеджер | Назначение исполнителей | Менеджер назначает исполнителя на задачу |
| Менеджер | Фиксация результатов | Менеджер фиксирует результаты выполнения работы |
| Исполнитель | Принять/оотклонить задачу | Позволяет принять задачу к исполнению или отказ от исполнения |
| Исполнитель | Исполнение | Отмечать задачу выполненной при завершении ее исполнения |

# Разработка диаграмм вариантов использования

Все варианты использования показаны на рис. 2.

A diagram of a person's flowchart

Description automatically generated with low confidence

**Рис. 2. Диаграмма прецедентов системы**

|  |
| --- |
| 1. **Краткое описание проекта** |
| *1.1 Название проекта* |
| *систему кконтроляя и учета выполнения задач и поручений* |
| *1.2* *Суть проекта* |
| *Суть проекта "система контроля и учета выполнения задач и поручений" заключается в создании программного обеспечения, которое позволит эффективно управлять и контролировать выполнение задач и поручений внутри компании или организации.* |
| 1.3 Бизнес-окружение проекта |
| повышение эффективности работы, улучшение качества продукции или услуг, оптимизация бизнес-процессов.   * Ожидания высшего менеджмента могут заключаться в улучшении управления процессами на предприятии, повышении производительности и качества работы сотрудников, сокращении времени на выполнение задач и поручений, увеличении точности и своевременности отчетности, улучшении взаимодействия между различными отделами и т.д. * Цели проекта связаны со стратегическими целями клиента в том смысле, что создание системы контроля и учета выполнения задач и поручений может помочь компании достичь своих более широких стратегических целей. Например, автоматизация управления проектами может помочь компании снизить затраты на управление проектами и повысить эффективность работы, что может увеличить ее конкурентоспособность и помочь ей выйти на новые рынки. * Целевая аудитория программного продукта может включать в себя различных пользователей, в том числе руководителей проектов, менеджеров, администраторов, исполнителей задач и других сотрудников компании. В зависимости от конкретных требований и потребностей компании, система контроля и учета выполнения задач и поручений может быть настроена для удовлетворения потребностей различных пользователей. |
| 1.4 Цели проекта и Требования по внедрению продукта |
| Цель проекта - разработка системы контроля выполнения задач и поручений, которая позволяет упростить и упростить управление задачами и поручениями, сократить время выполнения и повысить эффективность работы*.* |
| 1.5 Аналоги продукта |
| На рынке существует множество аналогов системы контроля и учета выполнения задач и поручений, некоторые из которых:  Trello – онлайн-сервис для управления задачами, который позволяет создавать доски и карточки для каждой задачи, назначать задачи на определенных пользователей, устанавливать сроки выполнения задач и т.д.  Asana – онлайн-сервис для управления задачами, который позволяет создавать проекты, задачи, подзадачи, назначать задачи на определенных пользователей, устанавливать сроки выполнения задач, создавать шаблоны задач и т.д.  JIRA – онлайн-сервис для управления проектами и задачами, который позволяет создавать задачи, устанавливать сроки выполнения задач, назначать задачи на определенных пользователей, отслеживать прогресс выполнения задач и т.д.  Microsoft Project – программное обеспечение для управления проектами, которое позволяет создавать графики выполнения задач, устанавливать сроки выполнения задач, отслеживать затраты на проект и т.д.  Basecamp – онлайн-сервис для управления проектами, который позволяет создавать проекты, задачи, документы, устанавливать сроки выполнения задач, назначать задачи на определенных пользователей и т.д.  Каждый из этих продуктов имеет свои уникальные функции и возможности, и выбор продукта зависит от конкретных потребностей и требований компании. |
| **2. Описание продукта и поставок** |
| *2.1 Продуктом проекта является (перечень поставок)* |
| * Система управления задачами – это основной компонент системы, который позволит пользователям создавать, назначать, отслеживать и управлять задачами и поручениями. * Модуль отчетности – компонент системы, который будет позволять формировать отчеты о выполнении задач и поручений. * Модуль уведомлений – компонент системы, который будет отслеживать сроки выполнения задач и поручений и автоматически отправлять уведомления пользователям о необходимости выполнения задач. * Модуль аналитики – компонент системы, который будет позволять анализировать данные о выполнении задач и поручений и выявлять тренды и паттерны. * Модуль интеграции – компонент системы, который будет позволять интегрировать систему с другими приложениями и сервисами, такими как электронная почта, календарь и т.д.   В зависимости от требований заказчика и конкретных потребностей проекта, список поставок может быть изменен или дополнен. |
| 2.2 Требованиям к продукту являются (продукт позволяет): |
| 2.2.1 Требования бизнес-уровня:   * Продукт должен обеспечивать эффективное управление задачами и поручениями в организации, что приведет к повышению производительности и улучшению бизнес-процессов. * Продукт должен быть готов к использованию в корпоративной среде и соответствовать требованиям безопасности информации.   2.2.2 Пользовательские требования (высокоуровневые):   * Продукт должен обладать простым и интуитивно понятным пользовательским интерфейсом. * Продукт должен позволять пользователям быстро создавать, назначать и управлять задачами и поручениями, а также легко получать доступ к информации об их статусе.   2.2.3 Требования системного уровня:   * Продукт должен быть готов к использованию в распределенной среде и поддерживать работу нескольких пользователей одновременно. * Продукт должен обеспечивать масштабируемость, чтобы можно было легко добавлять новых пользователей и расширять функциональность системы.   2.2.4 Требования технического уровня:   * Продукт должен быть разработан с использованием современных технологий и языков программирования. * Продукт должен обеспечивать высокую производительность и надежность, а также быть готовым к работе с большим объемом данных. |
| 2.3 Требованиями к продукту НЕ являются (продукт не включает): |
| * Функциональность, не связанная с контролем и учетом выполнения задач и поручений. * Интерфейс и дизайн, не являющиеся критическими для работы продукта. * Функциональность, которая не была оговорена в требованиях к продукту. * Любые виды содержимого, которые не относятся к задачам и поручениям, такие как изображения, видео и звуки. * Функциональность, не поддерживающая необходимую производительность и надежность продукта. |
| *2.4 Правила приемки поставок:* |
| * Комиссия приемки: Обычно создается комиссия для проведения приемки, состоящая из представителей заказчика, разработчиков и экспертов. Комиссия оценивает соответствие продукта требованиям, проводит тестирование и анализирует отчеты о выполненной работе. * Критерии приемки: Определение критериев приемки должно быть оговорено на начальных этапах проекта. Они могут включать проверку функциональности, соответствие требованиям, производительность, безопасность и другие характеристики продукта. * Документы для сдачи: Сдача продукта должна осуществляться на основании утвержденных документов, таких как план проекта, требования к продукту, отчеты о тестировании и документация о продукте. Предоставление полной документации является обязательным условием приемки продукта. |

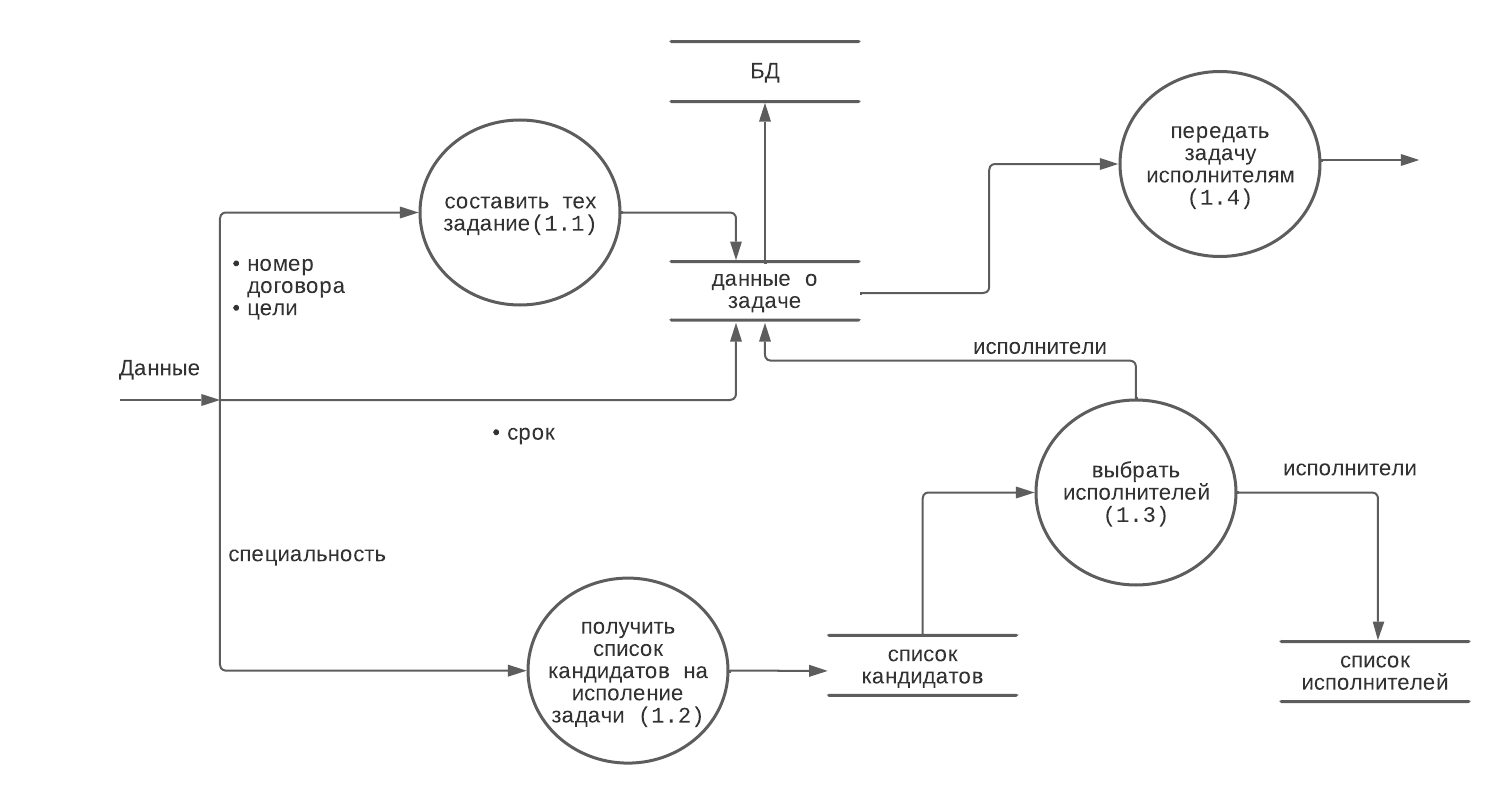
|  |  |
| --- | --- |
| **3. Ограничения проекта** | |
| 3.1 Вехи и дата завершения проекта: | |
| Начало проекта | *30.03.2023* |
| * Регистрация и вход с помощью логин и пароля | *01.04.2023* |
| * Создание и поставление задач | *10.04.2023* |
| * Выполнение и отправки задач на проверку | *20.04.23* |
| * Принимать или отправить на переделывание | *30.04.23* |
| Завершение проекта | *10.05.2023* |
| 3.2 Общий бюджет проекта | |
| Бюджет проекта отсутствует. Разработчики проекта используют личные ноутбуки и бесплатное программное обеспечение для разработки проекта. | |
| 3.3 Ограничения по выполнению и организации работ | |
| Технические ограничения:  • Приложение должно быть реализовано на языке Dart с  использованием технологии Flutter Framework  • В качестве СУБД должна быть использована SQLite  • Приложение должно работать на платформах Android и IOS  Ограничения по организации работ:  • Команда проекта состоит из двух разработчиков  • Для работы над проектом необходимо использовать собственное  оборудование и программное обеспечение  • Ограниченный доступ к серверным и клиентским ресурсам для  тестирования и разработки. | |
|  | |

|  |
| --- |
| **4. Руководитель проекта и его полномочия** |
| Мохамед М.О |
| 4.2 Полномочия руководителя проекта |
| Назначенный руководитель проекта будет отвечать за организацию и координацию работы команды разработки, обеспечивая успешное выполнение проекта в соответствии с поставленными задачами и сроками.   * Определение целей проекта и установление приоритетов в разработке, в тесном сотрудничестве с другим участником команды. * Разработка плана проекта, включая определение этапов и оценку временных рамок, и распределение задач между членами команды. * Определение требований к проекту и контроль за их выполнением. * Определение и установление процессов разработки и контроль за их выполнением, в том числе разработка методологии разработки, определение ролей и обязанностей каждого участника команды. * Оценка рисков проекта и разработка плана по их управлению. * Мониторинг и отчетность о состоянии проекта, включая подготовку и представление отчетов о прогрессе проекта, статусе выполнения   работ. |

|  |
| --- |
| **5. Заинтересованные лица и ресурсы** |
| 5.1 Заказчик проекта |
| Заказчик проекта может быть внутренним или внешним для компании, которая разрабатывает систему. Внутренний заказчик может быть руководителем отдела или департамента, который нуждается в такой системе для более эффективного управления задачами и поручениями внутри компании. Внешний заказчик может быть клиентом компании или другой организацией, которая хочет использовать систему для контроля выполнения задач, связанных с проектами или услугами, которые предоставляет компания. |
| 5.2 Ключевые пользователи результатов проекта: |
| * Руководители, которые используют систему для отслеживания и управления выполнением задач и поручений своих подчиненных, а также для принятия решений на основе собранных данных. * Сотрудники, которые будут использовать систему для учета выполнения своих задач и поручений, мониторинга своей продуктивности и получения обратной связи от своих руководителей. * IT-специалисты, которые будут устанавливать, настраивать и обслуживать систему, а также обеспечивать безопасность и защиту данных. * Владельцы бизнеса и высшее руководство, которые будут использовать данные, собранные системой, для принятия стратегических решений и определения направлений дальнейшего развития компании. * Заказчики, если система контроля и учета выполнения задач и поручений является частью услуги или продукта, предоставляемых компанией. * Регуляторные органы, если система используется в отраслях, где существуют строгие правила и нормативы, например, в фармацевтической или финансовой отраслях. * Внешние консультанты, которые могут использовать данные, собранные системой, для оценки эффективности и производительности компании. |
| 5.3 Спонсор проекта |
| Спонсор проекта может быть как внутренним, так и внешним для компании, которая разрабатывает систему. Внутренний спонсор может быть высшим руководством компании или руководителем отдела, который хочет получить финансирование для проекта. Внешний спонсор может быть клиентом компании или другой организацией, которая хочет внедрить такую систему в своей компании. |
| 5.4 Куратор проекта |
| Куратор проекта может быть представителем заказчика |
| 5.5 Команда проекта |
| Команда проекта состоит из человека, которые разрабатывают клиент - серверное приложение на Firebase. Они могут взаимодействовать с заказчиком, спонсором и куратором проекта, чтобы получить обратную связь и дополнительную поддержку. |
| 5.6 Инфраструктура |
| Для разработки клиент - серверного приложения на Firebase может потребоваться выделение определенных ресурсов, таких как сервера, базы данных и различное программное обеспечение. |

## Диаграммы потоков данных

***Рис.*** *5.2.1.* Контекстная диаграмма

Для добавления задачи Директору необходимо указать всю информации об этой задаче (срок, специальность, номер договора срок, цели,  и т.д.). Вся эта информация поступает менеджерам и обрабатывается. Процесс обработки задачи может быть детализирован DFD первого уровня как показано на рис. 5.2.2 

**Рис.** 5.2.2. Детализация процесса обработать задачу

**Процесс 1.1** (Составить техническое задание) осуществляет приём данных о целях поставленной задачи. На входе/выходе имеет следующие потоки:

* Входной поток “Цели” как элемент внешнего потока “Данные”
* Выходной поток - непосредственно “Техническое Задание”, которая заносится в поле Данных о задаче

**Процесс 1.2** (Получить кандидатов на исполнение задачи) выбирает из списка, предоставляемого базой данных исполнителей компании, людей, которые компетентны в специальности поставленной задачи. На входе/выходе имеет следующие потоки:

* Входной поток “Специальность”, формируемый посредством БД. Этот поток является элементом внешнего потока “Данные”
* Выходной поток “Список кандидатов”, который поступает в хранилище

**Процесс 1.3** (Выбор исполнителей) выбирает из списка кандидатов на исполнение задачи, сотрудников, которые удовлетворяют требованиям к исполнению задачи, например имеют достаточно свободных часов нагрузки. На входе/выходе имеет следующие потоки:

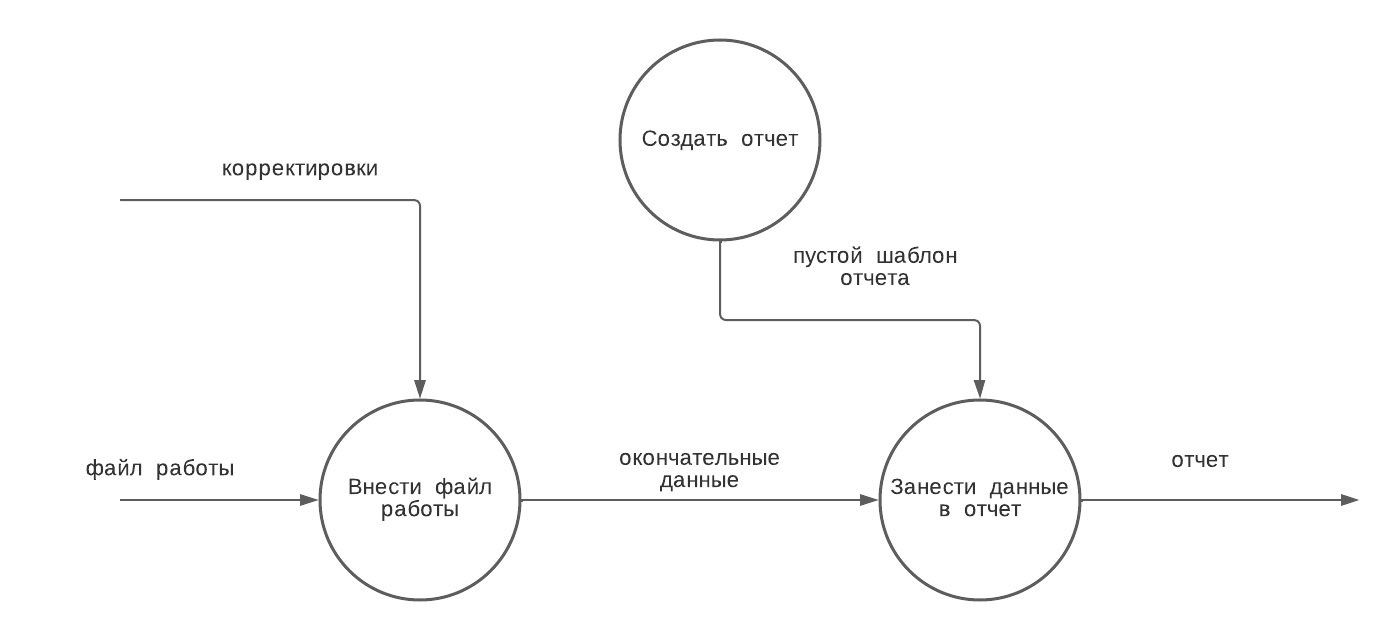
* Входной поток “Список Кандидатов” из хранилища участников
* Выходной поток “Список Исполнителей”, который поступает в хранилище

**Процесс 1.4** (Передача Задачи Исполнителям) Передает необходимую информацию и данные о задаче исполнителям. На входе/выходе имеет следующие потоки:

* Входной поток “Данные о Задаче”, который является совокупностью всех данных о задаче. Именно информацию из этого потока получают исполнители.
* Выходной поток “Техническое задание”, который отправляется исполнителям.

После выполнения задачи исполнителями формируется отчет о проделанной работе, после этот отчет проверяется Менеджером и отправляется Директору.

Процесс Формирования отчета может быть детализирован DFD первого уровня как показано на рис. 5.2.4



**Рис.** 5.2.4. Детализация процесса Формирование протокола заседания

**Процесс 2.1** (Внести файл работы) Добаляется конечный файл над которым работали исполнители или фото видео отчет. На входе/выходе имеет следующие потоки:

* Входной поток “Файл Работы ” как элемент внешнего потока
* Выходной поток “ Окончательные Данные”, которые заносятся в шаблон протокола

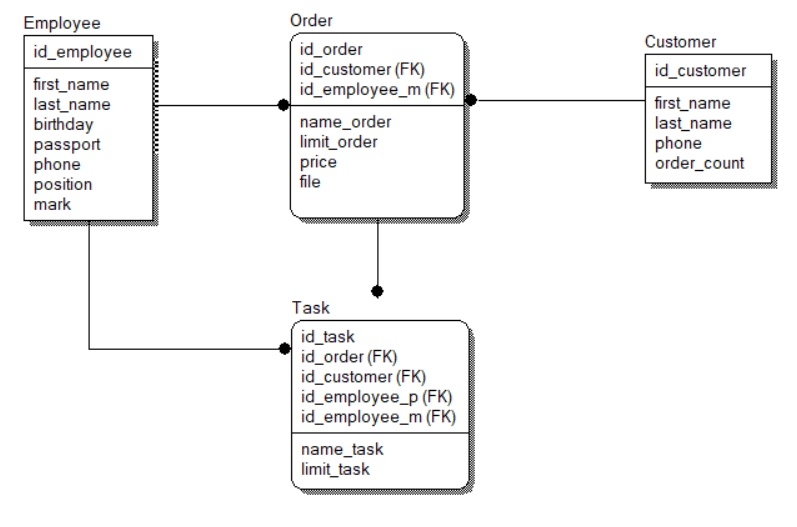
**Процесс 2.2.2** (Создать Отчет) формирует пустой шаблон отчета. На выходе имеет следующий поток:

* Выходной поток “Пустой шаблон”, в который заносятся данные о задаче

**Процесс 2.2.3** (Занести данные в шаблон) заносит скорректированные данные в шаблон и формирует окончательный отчет. На входе/выходе имеет следующие потоки:

* Входной поток “Окончательные данные”
* Входной поток “Пустой шаблон”
* Выходной поток “Отчет”, который передаётся далее

Диаграммы сущность-связь



**Рис.** 5.3.1.ER-диаграмма (сущность-связь)

**SADT Диаграммы**

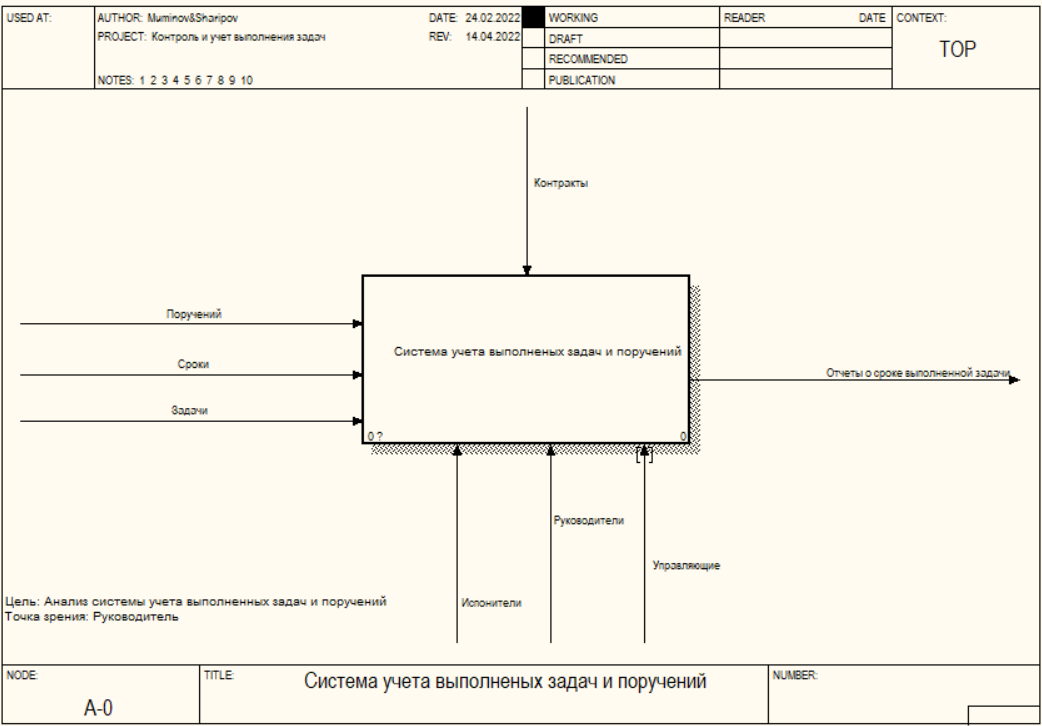


Рис. 1 Контекстная диаграмма

Таблица 1. Стрелки контекстной диаграммы

|  |  |  |
| --- | --- | --- |
| *Имя стрелки (Arrow Name)* | *Определение стрелки (Arrow Definition)* | *Тип стрелки (Arrow Type)* |
| Поручений | Предоставления поручений со стороны клиента | Input |
| Сроки | Предоставления сроки со стороны клиента | Input |
| Задачи | Предоставления задачи и ее условии | Input |
| Контракты | Договор цены за выполнении задачи, договор сроки исходя из сложности задачи, проверки и получения выполненной работы | Control |
| Руководитель | Руководитель программы (компании) | Mechanism |
| Управляющий | Управляющий проекта | Mechanism |
| Исполнитель | Исполнитель задач | Mechanism |
| Отчёт | Отчёт о выполненной задачи | Output |

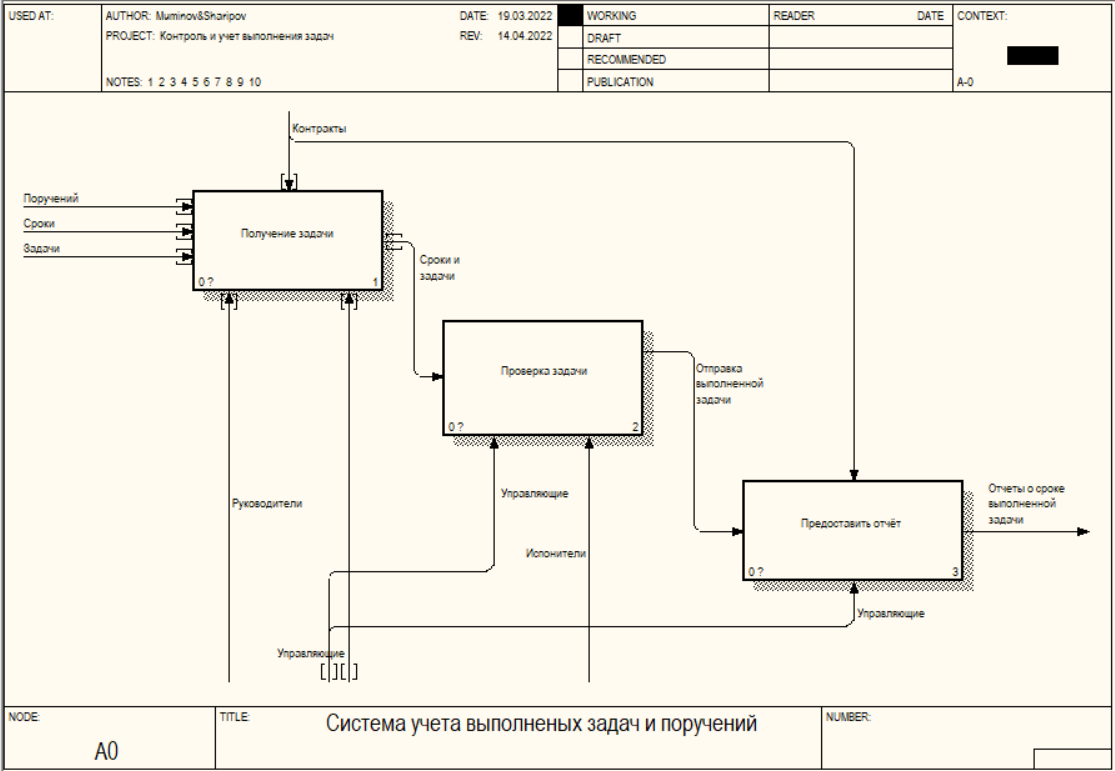


Рис. 2 Декомпозиция контекстной диаграммы.

Таблица 2*. Работы диаграммы декомпозиции А0*

|  |  |
| --- | --- |
| ***Имя работы (Activity Name)*** | ***Определение (Definition)*** |
| Получение задачи | Получить задачи, обсуждать сроки с заказчиком |
| Проверка задачи | Разбить задачи на подзадачи и следить за их выполнениями |
| Предоставить отчёт | Написать отчёт о выполненных задач и предоставить заказчику |

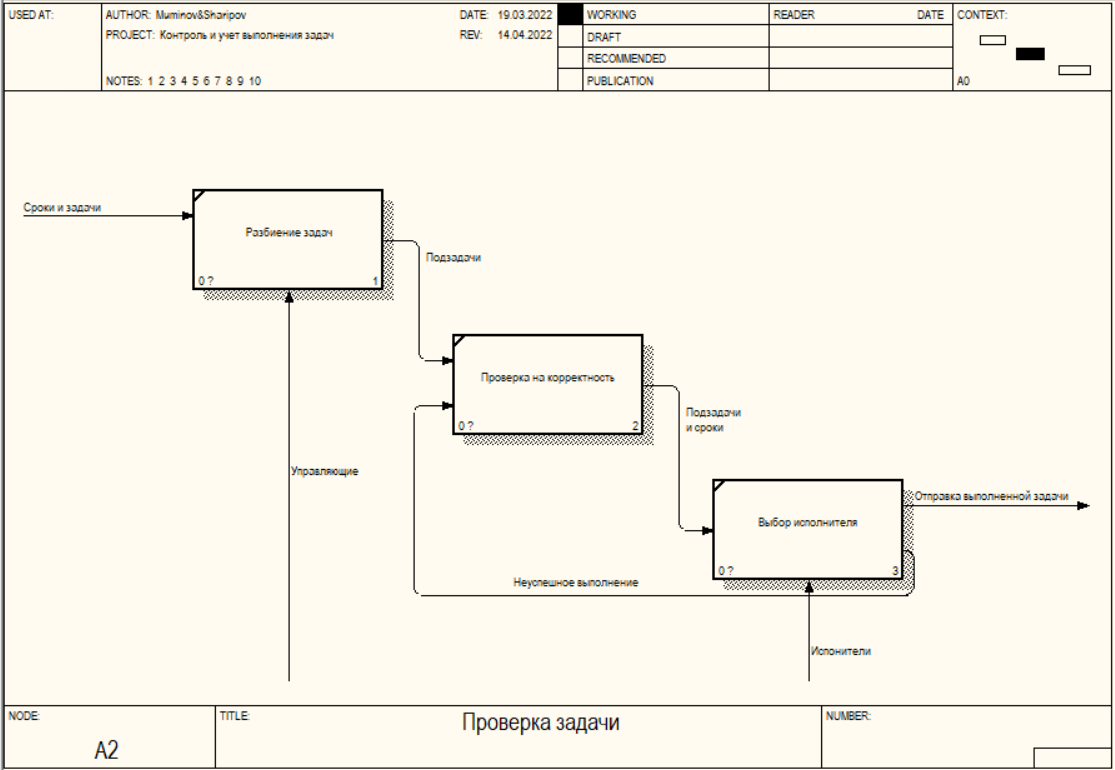


Рис. 3 Декомпозиция «Проверка задачи».

Таблица 3*. Работы диаграммы декомпозиции А2*

|  |  |
| --- | --- |
| *Имя работы (Activity)* | *Определение работы (Activity Definition)* |
| Разбиение задач | Разбиение задач на подзадачи |
| Проверка на корректность | Проверка на корректность подзадач |
| Выбор исполнителя | Выбрать исполнителя исходя из его опыта и свободного времени |

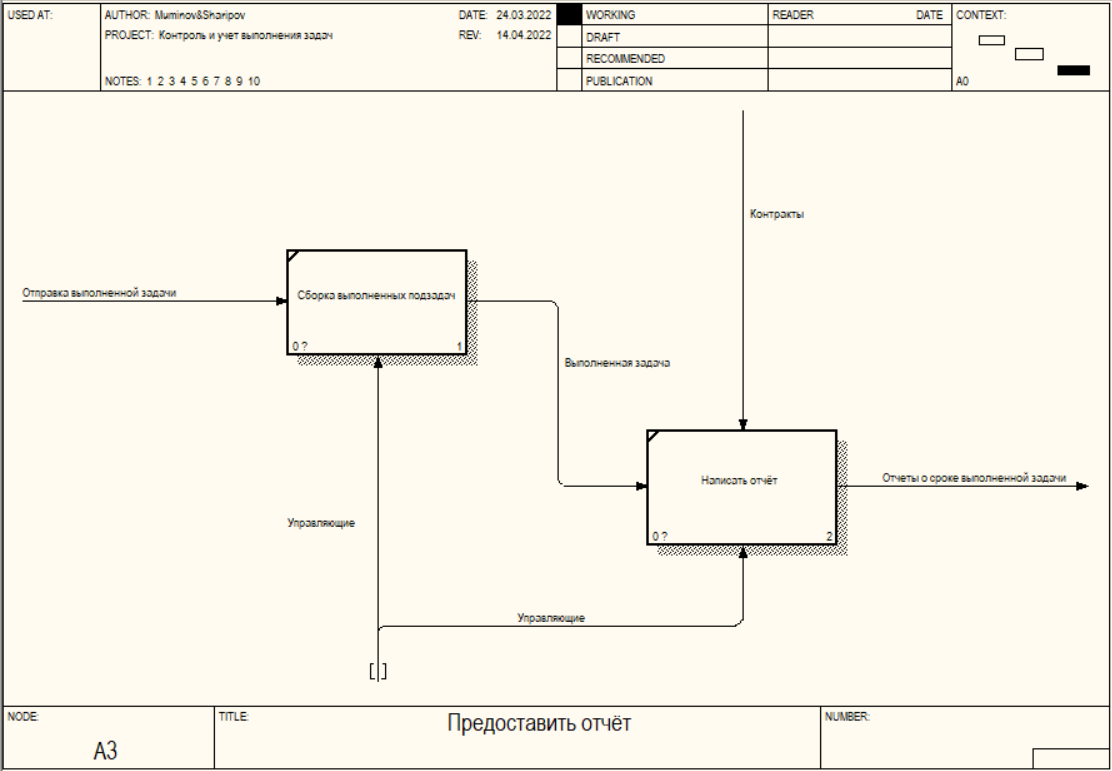


Рис. 4 Декомпозиция «Предоставить отчет».

Таблица 4*. Работы диаграммы декомпозиции А3*

|  |  |
| --- | --- |
| *Имя работы ( Activity Name)* | *Определение работы (Activity Definition)* |
| Сборка выполненных подзадач | Собрать всех выполненных подзадач |
| Написать отчёт | Написать отчёт о сроки выполненной задачи |

A picture containing diagram, plan, line, technical drawing

Description automatically generated5.2 Диаграмма классов предметной области

***Рис.*** *5.5.1.* Диаграмма классов

* 1. Архитектура приложения клиент-сервер

В нашей программной системе используется архитектура клиент-сервер

В программной системе "Система контроля и учета выполнения задач и поручений" используется архитектура клиент-сервер, которая обеспечивает эффективное взаимодействие между клиентскими приложениями и серверами. Архитектура клиент-сервер разделена на несколько уровней, каждый из которых выполняет определенные функции:

1. Нижний уровень - клиентские приложения:

* - Приложения клиентов: Расположены на компьютерах пользователей и выполняют функции и логику представлений. Клиентские приложения предоставляют программный интерфейс для вызова приложения на среднем уровне. В данном случае, в качестве клиента выступает приложение на смартфоне. Это так называемые "тонкие" или "облегченные" клиенты, которые осуществляют интерфейс пользователя с системой.

1. Средний уровень - сервер приложений:

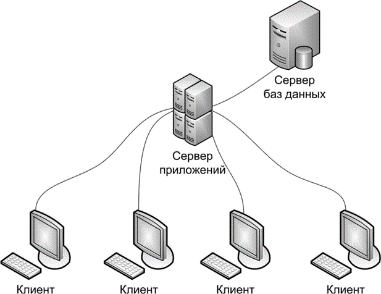
* - Сервер приложений: Расположен на сервере и выполняет прикладную логику системы. Здесь выполняются операции обработки данных и обмена информацией с базами данных. Сервер приложений является промежуточным звеном между клиентскими приложениями и базами данных. Он обеспечивает обмен данными между пользователями и базами данных, а также выполнение прикладной логики системы.

1. Верхний уровень - удаленный сервер базы данных:

* - Сервер базы данных: Расположен на отдельном сервере и принимает информацию от сервера приложений. Этот сервер базы данных специализирован для обработки данных и выполнения файловых операций. Он обеспечивает хранение и управление данными, связанными с выполнением задач и поручений. СУБД и база данных представлены в виде набора файлов на жестком диске специально выделенного компьютера с непрерывным доступом к сети Интернет.

В такой архитектуре клиенты инициируют обращение к программному обеспечению, расположенному на сервере приложений. Сервер приложений формирует запросы к базе данных и передает их по сети. Результат выполнения запроса копируется на сервер приложений, который затем возвращает результат клиентскому приложению для отображения пользователю.

Такая архитектура обеспечивает эффективное взаимодействие между клиентами и серверами, обеспечивает быстрый доступ к данным, целостность информации и обмен информацией между различными уровнями системы.



6.2. Структура программы с описанием функций составных частей   
и связи между ними

1. Клиентское приложение:

* Подсистема ввода данных: Обеспечивает пользовательский интерфейс для ввода информации о задачах и поручениях. Пользователи могут создавать новые задачи, указывать исполнителей, сроки выполнения и другие атрибуты задачи.
* Подсистема изменения данных: Позволяет пользователям вносить изменения в существующие задачи, такие как обновление статуса задачи, добавление комментариев, прикрепление файлов и т. д.

1. Сервер баз данных (FireBase):

* Хранение данных: Сервер баз данных FireBase обеспечивает хранение всех данных о задачах и поручениях. Он использует облачную СУБД класса NoSQL, которая позволяет эффективно хранить и синхронизировать данные между различными клиентами. Это обеспечивает доступность данных и согласованность информации для всех пользователей системы.

1. Взаимодействие клиентского приложения и сервера баз данных:

* Клиент-серверная связь: Клиентское приложение обращается к серверу баз данных для получения и отправки данных о задачах и поручениях. Это осуществляется через сетевое соединение, обычно по протоколу HTTP или HTTPS.
* Генерация команд: Клиентское приложение генерирует команды, основываясь на взаимодействии пользователя с интерфейсом приложения. Команды могут быть связаны с созданием, изменением или получением данных о задачах и поручениях.
* Автоматическая обработка команд: Сервер баз данных автоматически обрабатывает полученные команды от клиентского приложения. Он выполняет операции чтения и записи данных в базу данных FireBase, обновляет информацию о задачах и поручениях согласно командам.

1. Взаимодействие с регистратором:

* Доступ к базе данных: Регистратор имеет возможность напрямую взаимодействовать с базой данных. Это позволяет регистратору вносить корректировки в расписание приема пациентов или другие аспекты системы без использования клиентского приложения.

В целом, структура программы "Система контроля и учета выполнения задач и поручений" включает клиентское приложение с подсистемами ввода и изменения данных, сервер баз данных FireBase для хранения данных и обмена информацией с клиентским приложением, а также возможность напрямую взаимодействовать с базой данных для регистратора. Эта структура обеспечивает эффективное управление задачами и поручениями, обеспечивая доступность и целостность данных для пользователей системы.



Рис. 18. Структура программного комплекса

* 1. Используемые технические средства

Программное решение может эксплуатироваться на любом персональном компьютере (ПК) или смартфоне, имеющем выход в Интернет. Для комфортной работы в диалоговом режиме необходимы экран дисплея, клавиатура и манипулятор типа "мышь", высокоскоростной Интернет.

* 1. Вызов и загрузка

Вызов и загрузка программы осуществляется посредством перехода на главную страницу.

# Описание программы

## Общие сведения

* *Обозначение и наименование программы*

Название программы - “ Система контроля и учета выполнения задач и поручений” (СКиУЗ). Обозначение программы представлено на рисунке



***Рис.****.* Обозначение программы

* *Языки программирования, на которых написана программа*

Dart — язык программирования используемый в комплекте средств разработки и фреймворка Flutter с открытым исходным кодом для создания мобильных приложений под Android и iOS, веб-приложений, а также настольных приложений под Windows, macOS и Linux

## Функциональное назначение

* *Классы решаемых задач*

1. Задача автоматизации  
   Система позволяет оптимизировать и автоматизировать процесс контроля и учета выполнения задач и поручений, путем цифровизации процесса
2. Задача планирования

Контроль и отслеживание текущих задач по исполнителям позволяет компании составлять чёткий план своей работы, не тратя на это много времени и ресурсов.

1. Задача хранения   
   За счёт Базы Данных вся информация о выполненных задачах будет храниться сколько угодно без дополнительных затрат. Также не возникнет проблем с поиском необходимой информации.

* *Назначение программы*

“Система контроля и учета выполнения задач и поручений ” позволяет оперативно отслеживать ход выполнения задач, корректировать цели и средства. Эта система предназначена для руководящего состава, позволяя упрощать контроля выполнения задач, осуществлять мониторинг текущей ситуации и получать всю необходимую им информацию без лишних затрат времени и ресурсов. А также для сотрудников для отслеживания приближающихся сроках сдачи задачи.

* *Сведения о функциональных ограничениях на применение*

“Система контроля и учета выполнения задач и поручений” может работать только с устройствами, имеющими доступ к сети Интернет.

## Описание логической структуры

*Алгоритм программы*6.2. Логическая структура программы и обработка команд:

Программа "Система контроля и учета выполнения задач и поручений" обрабатывает команды, получаемые через интерфейс приложения на смартфоне. Интерфейсный модуль построен на двух типах диалогов: диалог "запрос – ответ" и диалог типа "меню". Взаимодействие между клиентом и сервером происходит следующим образом:

1. Запуск программы:

* - При запуске программы на смартфоне инициализируются все необходимые компоненты и модули для работы системы.

1. Ожидание нажатия кнопки:

* - После запуска, приложение ожидает нажатия кнопки на интерфейсе пользователем. Нажатие кнопки генерирует команду для обработки.

1. Обработка команды:

* - После получения команды от клиента, сервер автоматически обрабатывает команду. Он анализирует тип команды и определяет необходимые действия для выполнения.

1. Обращение к базе данных:

* - Для обмена информацией и получения необходимых данных, сервер обращается к базе данных. Он выполняет запросы к базе данных для получения требуемой информации или для обновления данных в соответствии с командой.

1. Отображение данных клиенту:

* - После обработки команды и получения необходимых данных из базы данных, сервер возвращает результат клиентскому приложению на смартфоне. Данные, полученные от сервера, отображаются на интерфейсе приложения для пользователя.

Таким образом, система обрабатывает команды, генерируемые пользователем через интерфейс приложения на смартфоне. Сервер автоматически обрабатывает эти команды, обращается к базе данных для обмена информацией и возвращает результаты клиентскому приложению для отображения пользователю.

Краткая блок-схема алгоритма работы программы изображена на рисунке

A picture containing diagram, text, sketch, plan

Description automatically generated8. Руководство пользователя

8.1 Назначение программы

Виды деятельности и функции, которые автоматизированные:

* Регистрация и вход с помощью логин и пароля;
* Создание и поставление задач;
* Выполнение и отправки задач на проверку;
* Принимать или отправить на переделывание.

Выполнение программы

1. Приветствие.



Рис.1

1. Если пользователь раньше не входил в систему то экран вход/регистация показывается. Вход с помощью электронной почты и пароля. (Рис.2 a) При нажатии на кнопку «Sign up» осуществляется переход на экран регистрации, где будучи сотрудник компании «ST Group» заполняет все необходимые данные для регистрации в системе и доступа к дальнейшим функциям. (Рис.2 b) При нажатии на кнопку «Sign in with Google» открывается окно выбора гугл аккаунта. (Рис.2 c) Если с этим аккаунтом не заполнены данные которые нужны компанию «ST Group», то переход на экран заполнения данные, иначе вход в систему. (Рис.2 d)

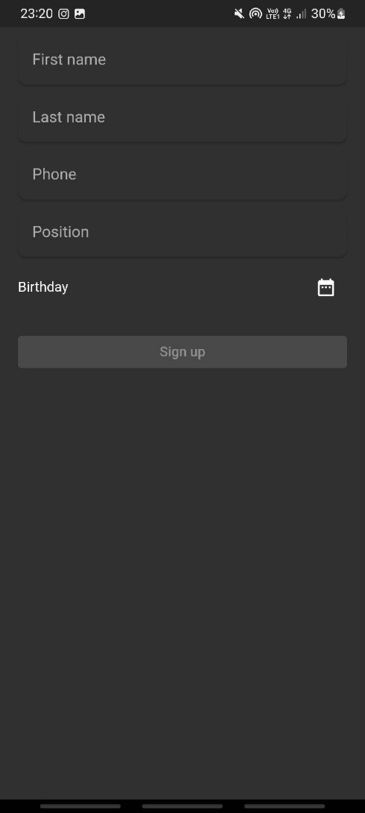
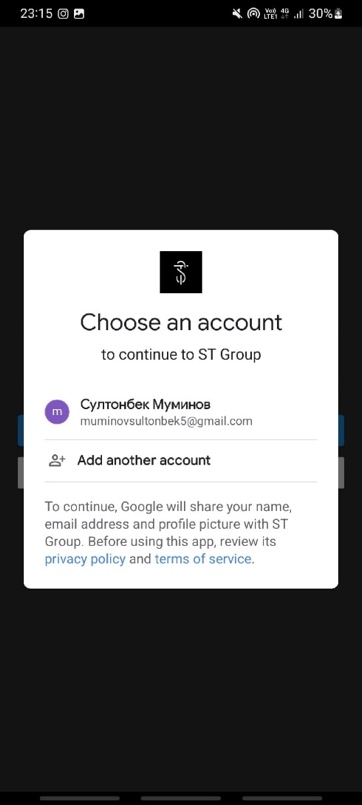
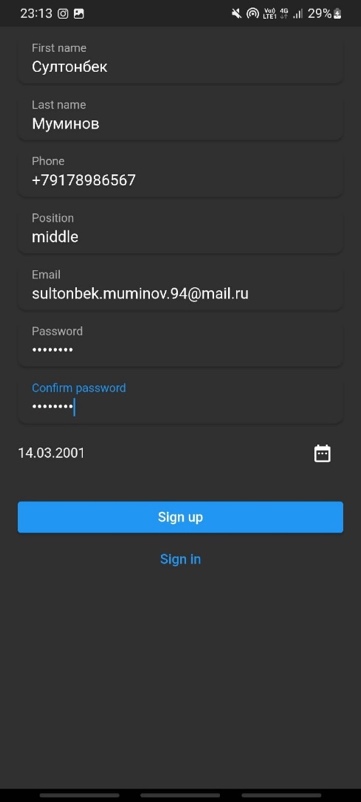
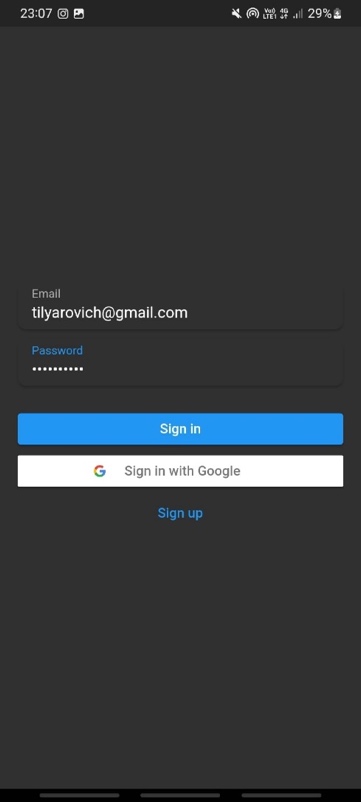


Рис.2 (a, b, c, d)

1. Если пользователь уже зарегистрирован то сразу открывается первый главный экран после приветствия. Это экран всех задач. В этом экране задачи расположены по приоритету (сначала те, которые мне поставили, потом те, которые я создал, а потом остальные). Иконки на задачах меняются, если меняется статус задачи. При нажатии на карточку задачи открывается экран где есть детальная информация о задаче. (Рис.4) При нажатии на кнопку «+» открывается экран создания новой задачи. (Рис.5) При нажатии на 2 или 3 иконку на нижней части экрана первый главный экран меняется на 2 или 3 главные экраны соответственно. (Рис.6, 7) Свайп карточки влево или вправо вызывает функцию удаления. Если у сотрудника есть достаточно прав на удаления задачи, то открывается окно и просить подтверждать действия, иначе сотрудник получает уведомление, о том что у него нет прав на удаления этой задачи.

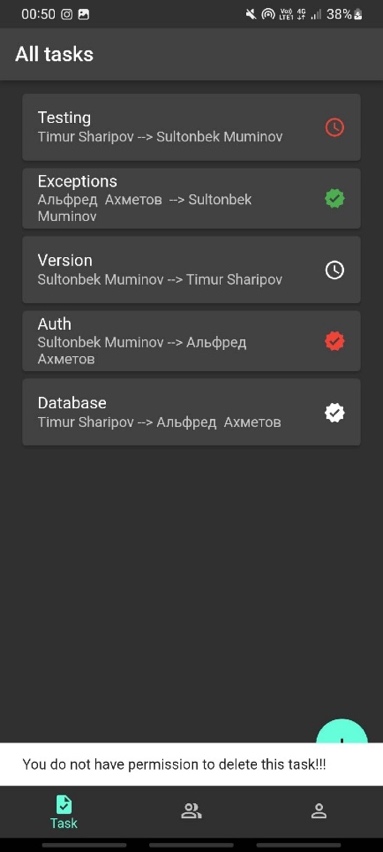
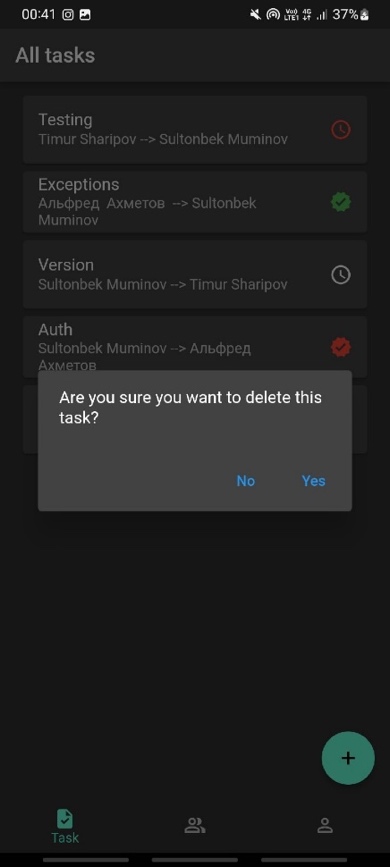
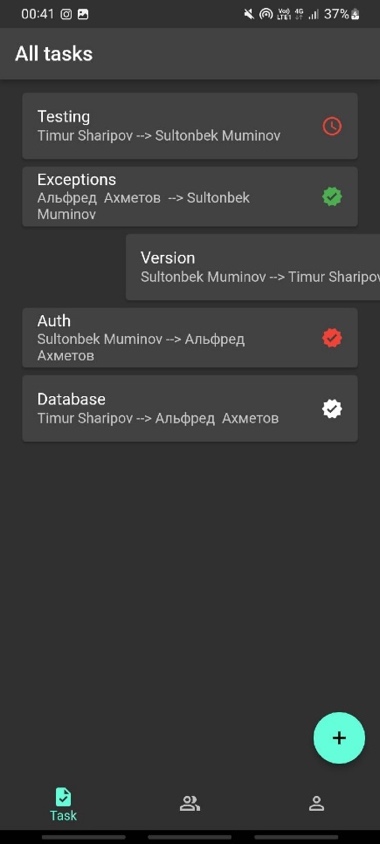
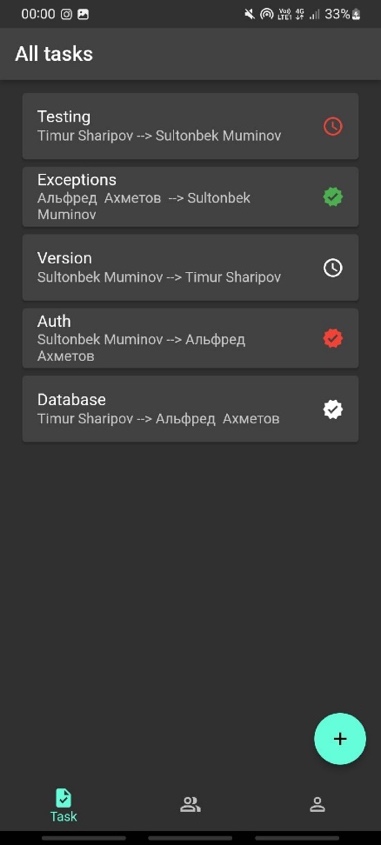


Рис.3

1. Экран детальной информации о задаче. Можно изменить задачу и нажатие на иконку, которая расположена на правом вверхнем углу отправить изменению в сервер. Этот экран выглядеть по разному, смотря какой статус задачи и кто является создателем или получателем задачи.

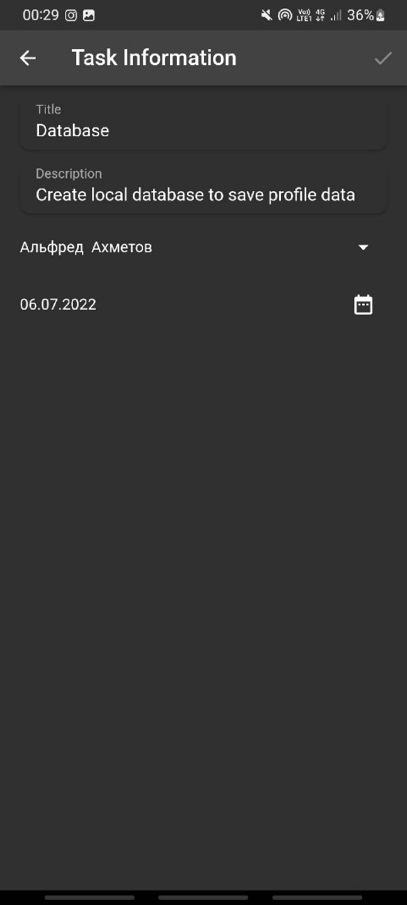
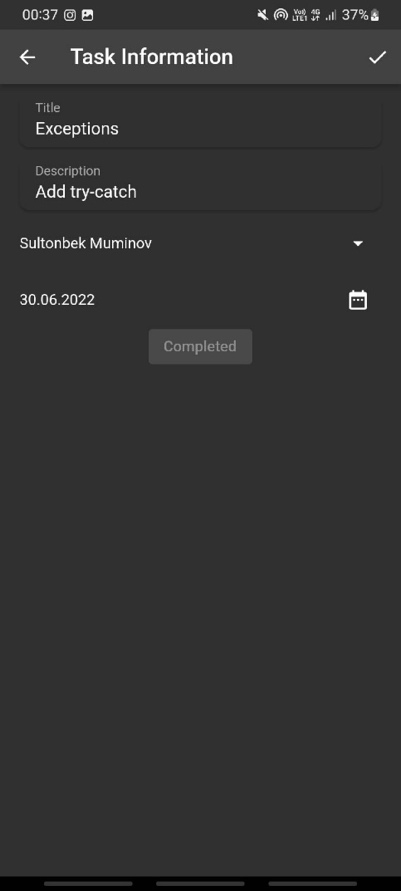
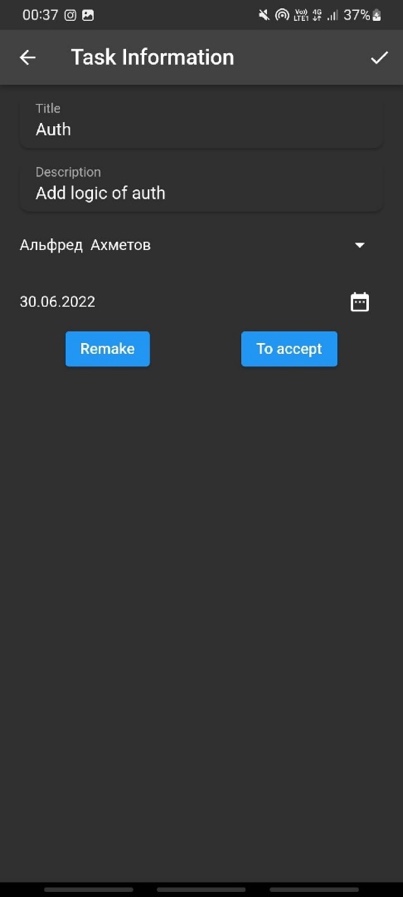
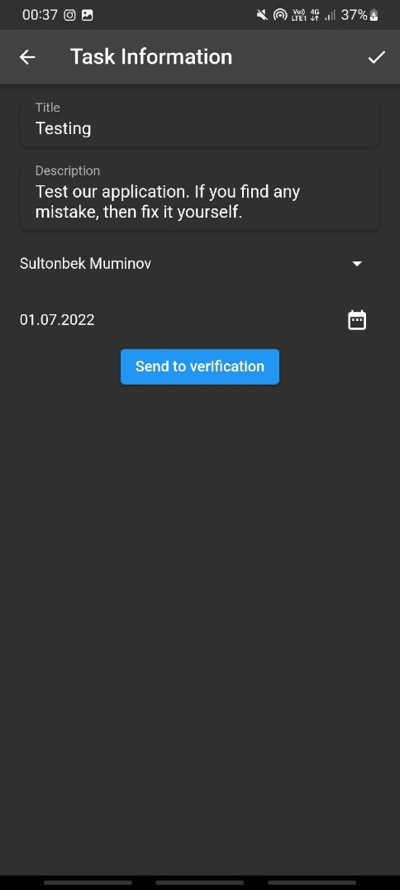


Рис.4

1. Экран список сотрудников. На этом экране можно увидеть список всех зарегистрированные сотрудники. При нажатии на каточку сотрудника карточка увеличивается и можно увидеть все данные этого сотрудника.

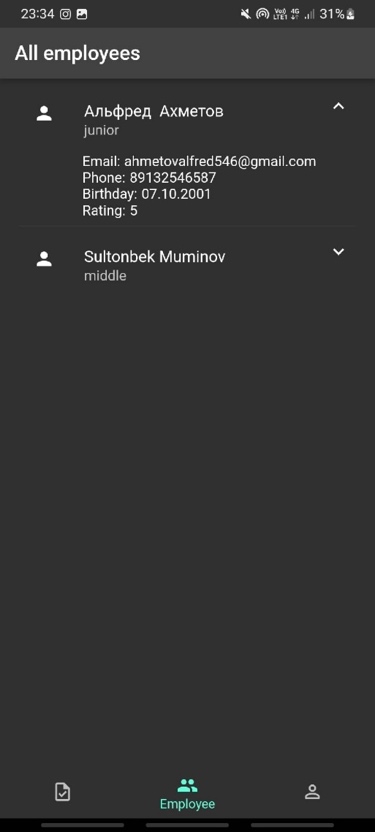
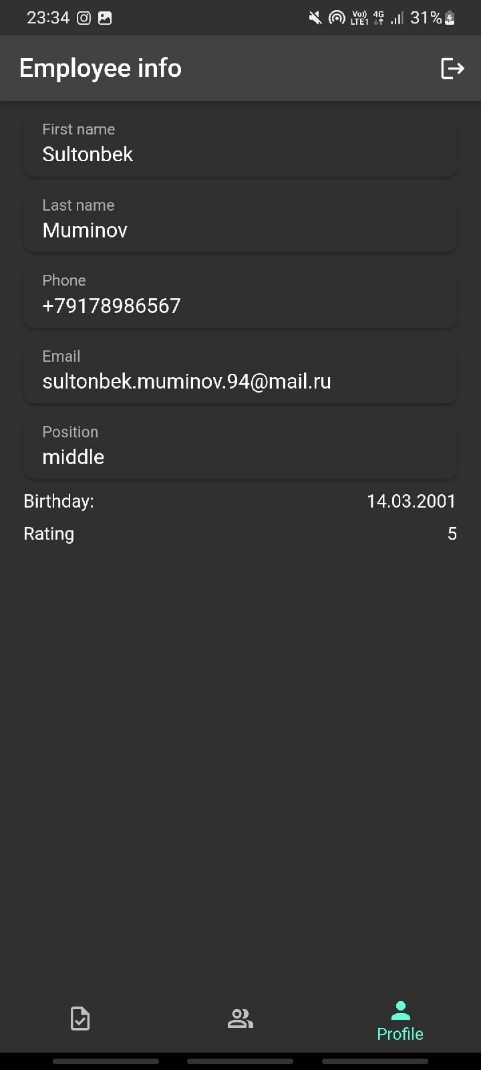
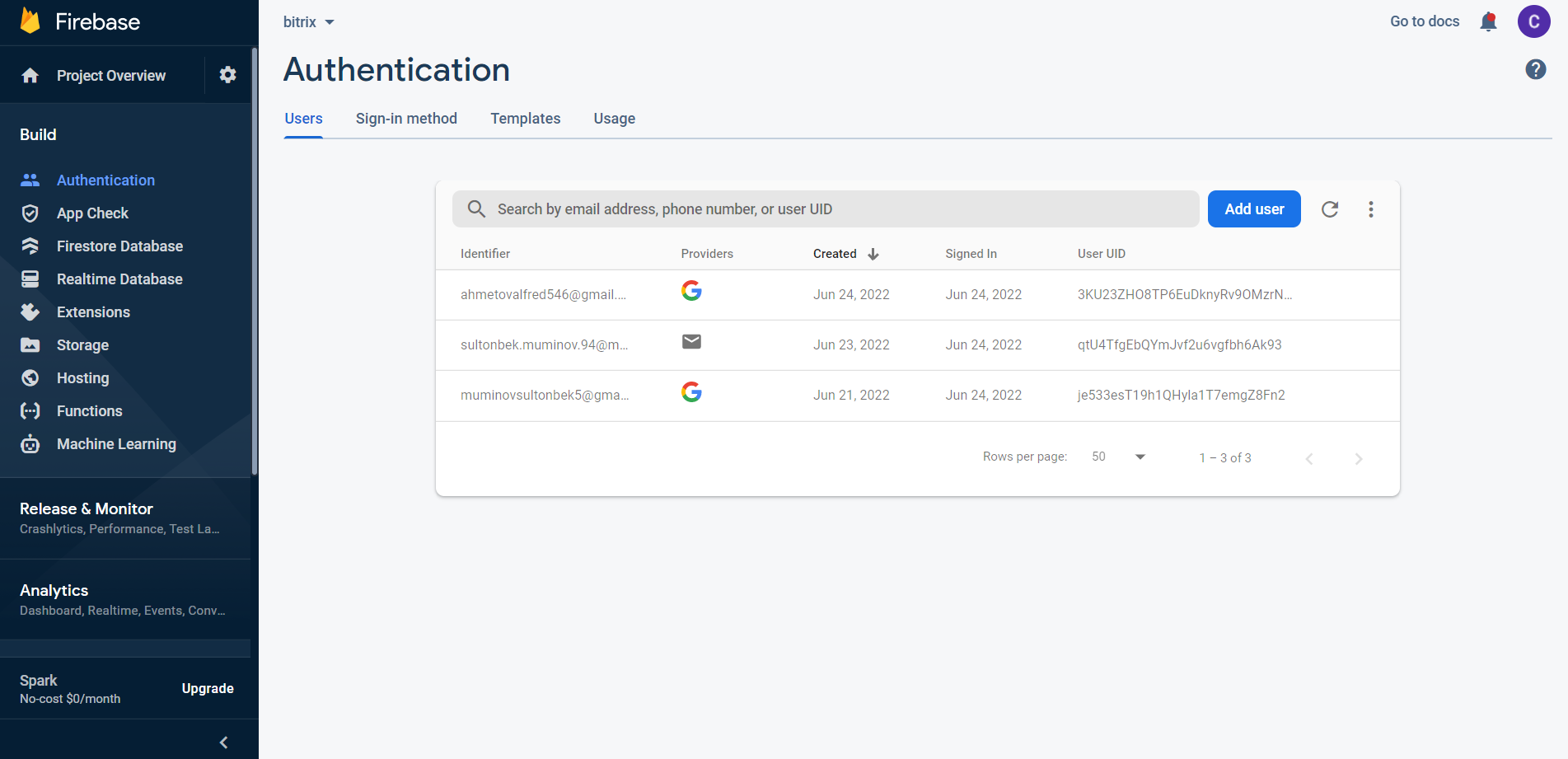


Рис.5

1. Экран профиль сотрудника. При нажатии на кнопку, которая расположена на правом вверхнем углу, происходить выход из системы.



8.2 Серверная часть:



Изображение выглядит как текст

Автоматически созданное описание

# Заключение

В рамках курсовой работы были приобретены и закреплены практические навыки в следующих видов:

Анализ предметной области и выявление основных требования к Системе контроля и учета выполнения задач и поручений.

* Разработка диаграмм вариантов использования, SADT диаграмм, диаграмм потоков данных, описывающих процессы, протекающих в системе, ER диаграмм, детализирующих необходимые структуры данных, и диаграмм классов предметной области, описывающих необходимые классы, методы и данные в разрабатываемой системе.
* Разработка связи с firebase
* Разработка клиентской части системы в виде пользовательского интерфейса
* Разработка руководства пользователя, подробно описывающего процесс использования разработанной системы.

# Глоссарий

Сайт, или веб-сайт, или Интернет-ресурс – это информационная единица в интернете, ресурс из веб-страниц (документов), которые объединены общей темой и связаны между друг с другом с помощью ссылок.

Браузер — прикладное программное обеспечение для просмотра страниц, содержания веб-документов, компьютерных файлов и их каталогов; управления веб-приложениями; а также для решения других задач.

Доменное имя – адрес веб-сайта. В данном случае это то, что отделено точкой от домена ru.

Сервер – место, где размещены файлы Вашего ресурса. Они хранятся не в облаках, а находятся физически на каком-либо устройстве, на котором предоставляет место хостинг.

База данных (БД) — это организованная структура, предназначенная для хранения, изменения и обработки взаимосвязанной информации, преимущественно больших объемов.

Аутентификация — проверка подлинности пользователя путём сравнения введённого им логина с паролем, сохранённым в базе данных пользовательских логинов.

Логин – это идентификатор пользователя для прохождения аутентификации в Интернет-ресурсе и входа в систему.

Пароль — условное слово или произвольный набор знаков, состоящий из букв, цифр и других символов, и предназначенный для подтверждения личности или полномочий.

Директор – руководитель компании, создает задачи.

Менеджер – управляет исполнителями и отчитывается перед Директором о проделанной работе. Более квалифицированный исполнитель.

Исполнитель – выполняет задачи поставленные директором через менеджеров и отчитывается перед ними.

Задача – задание поставленное Директором. Обязательно к исполнению.

Листинг

Main.Dart

import 'package:bitrix/data/repository.dart';

import 'package:bitrix/presentation/main\_scope.dart';

import 'package:firebase\_core/firebase\_core.dart';

import 'package:flutter/material.dart';

import 'firebase\_options.dart';

Future<void> main() async {

WidgetsFlutterBinding.ensureInitialized();

await Firebase.initializeApp(options: DefaultFirebaseOptions.currentPlatform);

await Repository().initHive();

runApp(const MyApp());

}

class MyApp extends StatelessWidget {

const MyApp({Key? key}) : super(key: key);

@override

Widget build(BuildContext context) {

return MaterialApp(

debugShowCheckedModeBanner: false,

title: 'ST Group',

theme: ThemeData.dark(),

home: const AuthScope(),

);

}

}