Вот как можно реализовать этот функционал, используя **Google Workspace (Таблицы, Диск, Скрипты, Почта)**, соблюдая требования к простоте и закрытому доступу:

**Основная концепция:** Одна главная Книга Google Таблиц с несколькими листами, доступ регулируется через Google Аккаунты сотрудников и скрипты.

**1. Структура Главной Книги Google Таблиц (например,**CRM\_Компания**):**

* **Лист: Менеджеры\_Заявки**
  + Столбцы как в п.1 ТЗ: Наим. орг., Контакт (Телефон, ФИО, Почта - можно 3 столбца), Товар, Цена, Ссылка на КП (в Диске), Ссылка на Договор/Спецификацию (в Диске), Ссылка на Заявку на прием (в Диске), Дата поставки, Дата послед. диалога, Статус (Выпадающий список: "Новая", "В работе", "Согласовано", "Отклонена" и т.д.), Заметки, Теги (можно один столбец, теги через запятую).
* **Лист: Логистика\_Прием**
  + Столбцы: Наим. орг., Ссылка на Заявку на прием (в Диске), Адрес склада, Контакт склада (Телефон, ФИО), Контакт водителя (ФИО, Телефон, Марка авто, Гос. номер - можно 4 столбца), Статус ("Ожидает", "В пути", "Принята", "Проблема"), Заметки, Теги.
* **Лист: Логистика\_Отправка**
  + Столбцы: Наш склад (Выпадающий список из справочника?), Склад покупателя, Контакт покупателя (Телефон, ФИО), Контакт водителя (ФИО, Телефон, Марка авто, Гос. номер), Статус ("Готовится", "В пути", "Доставлена", "Проблема"), Заметки, Теги.
* **Лист: Бухгалтерия\_Продажи (для ГенДира)**
  + Столбцы: Наим. товара, Цена поставщика (можно ссылаться на Менеджеры\_Заявки), Дата поставки, Дата продажи, Сумма продажи.
  + **Виджет суммы:** В отдельной ячейке (например, K1) формула =SUM(E:E) (если Сумма продажи в столбце E). Подписать "Общая сумма продаж:". Скрипт можно настроить на автоматическое обновление.
* **Лист: Бухгалтерия\_Расходы (для ГенДира)**
  + Столбцы: Название поставки, Статья расходов, Сумма расходов.
  + **Виджет суммы:** Аналогично, =SUM(C:C) (если Сумма расходов в C). "Общая сумма расходов:".
* **Лист: Зарплата (для ГенДира)**
  + Столбцы: Сотрудник (ФИО), Поставка (связь с Логистика\_Прием/Отправка или Бухгалтерия\_Продажи), Цена продажи (можно ссылаться), % вознаграждения, Сумма к выплате (формула =Цена продажи \* % / 100), Статус ("Рассчитана", "Выплачена").
* **Лист: Справочник\_Теги (для ГенДира)**
  + Столбцы: Тип ("Менеджер", "Логист", "Прием", "Отправка" и т.д.), Значение тега. Используется для создания выпадающих списков в других листах через Data Validation -> List from a range.
* **Лист: Справочник\_Склады (опционально)**
* **Лист: Справочник\_Сотрудники (для ГенДира)**
  + Столбцы: ФИО, Должность ("Менеджер", "Логист", "Проверяющий", "Админ"), Email (Google Аккаунт), Статус ("Активен", "Заблокирован").

**2. Google Диск: Файлообменник и Хранение Вложений**

* Создать структуру папок:
  + CRM\_Компания\_Документы
    - /Менеджеры/КП
    - /Менеджеры/Договоры\_Спецификации
    - /Менеджеры/Заявки\_на\_прием
    - /Логистика/Заявки\_на\_прием (можно ссылаться на те же, что у менеджеров)
    - /Логистика/Документы\_поставок
    - /Бухгалтерия
    - /Общие\_папки/[Название проекта] (для межотдельного взаимодействия)
* **В столбцах "Ссылка на ..."** в таблицах будет **ссылка** на соответствующий файл на Диске. Использовать функцию =HYPERLINK("URL\_файла"; "Название файла").
* **Для фото/документов/видео:** В последнем столбце на листах Менеджера и Логистики (напр., "Медиа") - **ссылка на папку на Google Диске**, созданную специально для этой конкретной заявки/поставки (например, CRM\_Компания\_Документы/Менеджеры/Заявка\_12345\_Медиа). В эту папку сотрудник загружает файлы (до 10 фото, до 5 док, до 2 видео). Название папки можно генерировать на основе ID строки.

**3. Реализация Функционала (Google Apps Script + Интеграции):**

* **Закрытый доступ:**
  + Гендир (Админ) создает Google Аккаунты для сотрудников (например, employee@yourdomain.com).
  + Настройка прав доступа к **Книге Таблиц** и **Папке на Диске**:
    - Админ (Гендир): Редактор на всё.
    - Менеджер: Редактор на лист Менеджеры\_Заявки и свои папки на Диске. Комментатор/Читатель на Логистика\_Прием/Отправка (если нужно).
    - Логист: Редактор на листы Логистика\_Прием, Логистика\_Отправка и свои папки на Диске.
    - Проверяющий: Читатель на листы Менеджеры\_Заявки, Логистика\_Прием, Логистика\_Отправка и соответствующие папки на Диске.
    - Бухгалтер (если будет отдельно): Читатель/Комментатор на Бухгалтерия\_\*.
  + **Скрипт:** Можно написать скрипт, который по роли сотрудника (из Справочник\_Сотрудники) автоматически настраивает права при добавлении строки с новым email. Или делать это вручную через интерфейс шаринга.
* **WhatsApp по API:**
  + Использовать внешние сервисы типа **Twilio API for WhatsApp** или **Green API**.
  + **Скрипт:** Написать обработчик в Google Apps Script, который:
    1. При изменении статуса заявки/поставки или даты последнего диалога (или по кнопке) берет номер телефона из таблицы.
    2. Отправляет запрос через выбранный API сервис для отправки уведомления/сообщения в WhatsApp.
    3. Логирует отправку.
  + Требует регистрации на внешнем сервисе, настройки вебхуков и оплаты.
* **Система уведомлений (Колокольчик):**
  + **Скрипт:** Создать отдельный маленький Лист: Уведомления со столбцами: Сотрудник (email), Текст уведомления, Ссылка на строку/объект, Время, Прочитано (Да/Нет).
  + Скрипт отслеживает изменения (например, новая заявка, изменение статуса, назначенная поставка) и добавляет строку в этот лист для нужного сотрудника.
  + Создать пользовательское меню или боковую панель в Таблицах с помощью HtmlService, которая:
    1. Определяет текущего пользователя (Session.getActiveUser().getEmail()).
    2. Читает лист Уведомления, фильтрует по email и Прочитано="Нет".
    3. Показывает число непрочитанных (колокольчик с цифрой) и список уведомлений. При клике помечает как прочитанное и открывает ссылку.
* **Редактор PDF:**
  + **Внешняя интеграция:** Использовать **Smallpdf API**, **iLovePDF API** или **Adobe PDF Services API**.
  + **Скрипт:** Добавить к столбцам с документами кнопки "Редактировать" (для PDF) и "Конвертировать" (для Word<->PDF).
  + При нажатии "Редактировать": Скрипт берет ссылку на PDF с Диска -> Отправляет во внешний API -> Получает ссылку на редактор -> Открывает ее в новом окне. После сохранения пользователем во внешнем редакторе, обновленный файл сохраняется обратно в Диск (часто через коллбэк вебхук).
  + При нажатии "Конвертировать": Скрипт отправляет файл на конвертацию через API и заменяет ссылку в таблице на новый сконвертированный файл.
  + Требует регистрации на внешнем сервисе и оплаты.
* **Взаимодействие между отделами (Поделиться строкой):**
  + **Скрипт:** Создать пользовательское меню "Действия" -> "Поделиться с отделом".
  + При выборе строки и вызове меню, скрипт показывает диалог с выбором отдела ("Логистика").
  + Скрипт копирует основные данные из текущей строки менеджера (Менеджеры\_Заявки) в новую строку на листе Логистика\_Прием (или Отправка), включая ссылки на документы и медиа-папку. Добавляет уведомление логисту.
* **Гибкие столбцы/теги (для ГенДира):**
  + **Справочник тегов:** Гендир редактирует лист Справочник\_Теги. Выпадающие списки в других листах автоматически обновляют опции.
  + **Добавление столбцов:** Гендир просто добавляет новый столбец на нужный лист вручную. Если нужна сложная логика, можно добавить скрипт для копирования структуры в "шаблонные" проекты.

**4. Интеграция с Почтой (доменная почта на reg.ru):**

* **Вариант 1 (Простой):** В боковой панели (через HtmlService) вставить веб-версию почтового клиента (например, Roundcube или стандартный интерфейс reg.ru) в iframe. Сотрудник логинится там отдельно. Минус: Не полноценная интеграция.
* **Вариант 2 (Gmail как прокси):** Настроить пересылку входящих писем с доменной почты на аккаунты Google сотрудников. Отправлять письма можно через Gmail, указывая доменный адрес как "отправлено от". Требует настройки SPF/DKIM.
* **Вариант 3 (API):** Использовать API почтового сервера (если reg.ru предоставляет) или стандартные протоколы (IMAP/SMTP) через Apps Script для чтения/отправки писем прямо из интерфейса CRM (боковая панель). Самый сложный в настройке.

**5. Роли (Админские функции):**

* **Создание/Блокировка аккаунтов:** Гендир управляет аккаунтами в Google Admin Console (создает, удаляет, приостанавливает пользователей employee@yourdomain.com).
* **Назначение ролей:** Гендир управляет правами доступа к Книге Таблиц и папкам на Диске через интерфейс шаринга Google. Роль сотрудника ("Менеджер", "Логист" и т.д.) хранится в Справочник\_Сотрудники и используется для отображения/логики в скриптах (например, какие уведомления показывать).

**Ключевые преимущества подхода:**

1. **Облачно:** Полностью в браузере, доступ откуда угодно.
2. **Простота (относительно):** Основа - знакомые Google Таблицы. Минимум сложных формул (основная логика в скриптах).
3. **Закрытый доступ:** Управление через Google Аккаунты и настройки прав.
4. **Интегрируемо:** Google Apps Script позволяет связать Таблицы, Диск, Почту и внешние API.
5. **Масштабируемо:** Можно постепенно добавлять скрипты и функционал.
6. **Стоимость:** Основные затраты - подписка Google Workspace (Business Standard/Plus для API квот) + возможные внешние API (WhatsApp, PDF).

**Сложности и что потребует разработки:**

* **Основная работа:** Написание и отладка Google Apps Script для:
  + Уведомлений (внутренних и WhatsApp).
  + Межотдельного взаимодействия (копирование строк).
  + Интеграции с внешними API (WhatsApp, PDF редакторы).
  + Пользовательского интерфейса (меню, боковая панель уведомлений).
* **Настройка API:** Регистрация, получение ключей, настройка вебхуков для Twilio/GreenAPI/Smallpdf и т.д.
* **Администрирование:** Настройка Google Аккаунтов, прав доступа, структуры Диска.
* **Производительность:** Таблицы могут тормозить при очень большом объеме данных. Нужна аккуратная структура и архивация.

**Следующие шаги:**

1. **Создать структуру:** Создать Книгу Таблиц с листами и структуру папок на Диске.
2. **Настроить аккаунты:** Завести Google Workspace, создать доменные аккаунты сотрудникам.
3. **Настроить базовые права:** Вручную выдать права на Книгу и папки Диска.
4. **Разработать скрипты:** Начать с самого критичного функционала (уведомления, копирование строк в логистику) или найти разработчика Google Apps Script.
5. **Интегрировать API:** Зарегистрироваться на нужных сервисах и интегрировать их со скриптами.

Этот подход дает максимально "примитивный" интерфейс (Таблицы) с возможностью наращивания функционала через скрипты и интеграции, полностью в облаке и с контролем доступа.