**Этап 1. MVP**

Требуется разработать программу, которая:

* Считывает из файлов информацию о Сотрудниках и их Задачах
* Генерирует различные отчеты по этим задачам в формате json
* Записывает в файлы эти отчеты

Рекомендуется разбить программу на несколько модулей, а именно:

* Модуль сущностей. Там следует описать сами модели данных
* Модуль работы с файлами
* Модуль логики формирования отчетов

**Модели**

В системе есть Сотрудники, выполняющие Задачи

* **Сотрудник** имеет
  + Id
  + Имя,
  + Должность (фронтендер, бэкендер, Аналитик, Тимлид, бухгалтер, Scrum-мастер, Администратор)
  + Непосредственного начальника (сотрудника)
* **Задача** имеет:
  + Id
  + Название,
  + Дату создания,
  + Дедлайн,
  + Исполнителя (сотрудника), Статус (запланирована, в разработке, на проверке, закрыта)
  + Риски (вероятность того, что задача не закроется в срок)
    - Gray – нет оценки
    - Green – низкий риск
    - Yellow – средний риск
    - Red – высокий риск
  + Подзадачи. Задача не может быть переведена в статус Закрыта, пока не закрыты все остальные подзадачи!
* Модели отчетов по задачам должны включать:
  + Для определенного сотрудника
    - Названия задач
    - Дедлайны задач
    - Риски задач
    - Статусы задач
  + Созданные за определенное время
    - Названия задач
    - Кол-во подзадач
    - Статусы задач
  + С определенным статусом
    - Названия задач
    - Имена сотрудников
    - Дедлайны задачи
  + С определенным риском
    - Названия задач
    - Кол-во подзадач
  + Для всех подзадач конкретной задачи
    - Названия задач
    - Дедлайны задач
    - Статусы задач

**Этап 2. Работа с сущностями**

Дополнить программу механизмом авторизации.

Добавить класс *CurrentConfiguration*, где будет храниться информация о пользователе, под которым вы вошли в систему.

При запуске программа должна запрашивать логин и пароль.

Если в системе находится пользователь с такими логином и паролем, информация об этом пользователе записывается в *CurrentConfiguration*

Добавить модуль для работы с задачами. Этот модуль должен предоставлять следующий функционал

* Всем категориям пользователей, кроме Бухгалтера:
  + Создавать задачу
  + Назначать исполнителем этой задачи либо себя, либо сотрудника, начальником которого этот пользователь является
  + Редактировать свою задачу следующим образом:
    - Менять ее статус. Обратите внимание, что сама Задача не должна позволять поставить статус Закрыта до тех пор, пока есть подзадачи НЕ в статусе Закрыта
    - Менять ее Риск
    - Добавлять новую подзадачу
  + Менять пароль всем сотрудникам, начальником которого он является
* Администратору:
  + Менять пароль любому сотруднику
  + Создавать задачу и назначать исполнителем кого угодно
  + Создавать новых сотрудников и назначать им Начальников

Так же требуется добавить в модуль для работы с файлами следующий функционал:

* Добавление новых сотрудников
* Добавление новых задач
* Обновление сотрудников
* Обновление задач

Обновленные данные требуется записывать в хранилище в том же формате, в котором они изначально там лежали

**Этап 3.**

1

**Этап 4.**

1