Генерация персоны.

Персона должна генерироваться и соответствовать следующему объектному представлению:

Person:

id: unique key for id of new Person: serial number + time stamp, devided “\_”,   
 name: “fake name”,

age: Random(16-24),

isHasPassport: Random(true / false),

gender: Random(“male” / ”female”),

payment: Random(900-1100$), step 100$,

healty: Random(65-95%),

Также должен быть конфигурационный файл для создания правил, согласно которым персоны будут генерироваться персон:

Configuration:

name: null / [“name1”, “name2”, ...“name5”], (min 2 persons, max 5 persons)

age: null / array of ages each person,

isHasPassport: null / array,

gender: null / array,

payment: null / array,,

healty: null / array,

quantityPersons: null / 2-5,

На UI должна быть форма заполнения данной конфигурации, соответственно если все поля равны null - тогда сетятся дефолтные значения. Если нет - тогда значения, взятые из конфигуратора (это форма с инпутами куда нужно прописывать все настройки если таковы будут необходимы).

После заполнения конфигурации должна раздезейблиться кнопка “GetVisaForAll” и после нажатия на неё - приложение стартует. Убрана логика генерирования персон каждые 30 секунд. Просто генерируем 2-5 персон и смотрим на результат их запроса на получение визы.

Должна быть отрисована вся иерархия прохода запроса по всем проверяющим инстансам. Как вариант, это квадраты со связями и с описанием каждого пункта проверки информации, а также цветового его заполнения в соответствии от состояния обещания: жёлтый - pending, красный - rejected, зелёный - resolved.

Основной пункт: Migration Service который валидирует имя и всё. Главное чтобы к нему не пришла одна и та же персона с одинаковым именем. То есть если я сам задам array одинаковых имён, то по сути только одна персона сможет получить визу.

У него есть три подпункта проверки (у которых, как и у персоны должен быть конфигурационный файл, который тоже должен присутствовать на ui части, то есть есть дефолтные правила которые описаны ниже, а вот я смогу задать и свои правила при необходимости):

1. Police Department:

* checking rule of age (age > 18, 5sec);
* checking rule of linking gender with age ( male: age must be more than 22, female: age must be more than 18, 8 sec)
* checking rule of having passport (12sec).

1. Medical Department:

* checking rule of healthy (healthy > 75%, 15sec);
* checking rule of linking gender with healthy ( male: healthy must be more than 75%, female: healthy must be more than 85%, 15sec)

1. Bank Department:

* checking rule of linking gender with payment (male: payment must be more than 1000$, female: payment must be more then 950$, 40sec)

Таким образом если по каждому инстансу вернётся resolve - тогда Migration Service выполнит обещание для одного из n пользователей, поскольку в данном случае необходимо будет использовать promise.race([promise1, promise2, ... , promiseN]). А для обращения каждого департамента в свои подпункты проверки информации необходимо воспользоваться promise.all([promise1, promise2, …, promiseN]).

валидация проходит на этапе заполнения полей конфигуратора то есть если ты ввёл 4 человека а в массиве только три имени - должна быть подсказка юзеру что это неправильно либо четко указываем конфигурацию либо поле пустое тоесть null

если получаем на какой из этапов проверки получения визы reject - мы юайно отображаем на каком из этапов это произошло (то есть где нам или в чём нам отказали) и на этом этапе у тебя в promise.race один из промисов вернёт reject - тоесть да, дальнейшая проверка не имеет никакого смысла если одна из инстансов или его подпунктов вернула reject

на ui части я вижу: 1) конфигуратор персон; 2) конфигуратор правил для всех инстансов; 3) кнопку старта; 4) основное поле отображения всей истории получения визы

воспринимайте каждый инстанс как callback функцию которую вы передаёте в then