# Instructions

This is a very open ended coding exercise. We are looking to understand how you go about designing and implementing software which will provide us insights into various aspects of your software engineering skills. The exercise does not require ORM or web application. It should be a pure in-memory OOP application.

1. Please describe what assumptions you are making about each of the classes/objects you are designing if any. You are welcome to create as many objects as needed.
2. Use TDD to implement and you can use any language and test framework that you are most comfortable with and any fixture creating libraries like factory boy and faker in python
3. Please share a public github repo with a good README with reasonable documentation as if you are handing it off to another engineer integrate and it will be great if you have github commit history vs just uploading the final finished software

# Programming Exercise

1. There are 1000 consumers
    - Consumers can make calls (assume they can only call to 1 number that will route to an agent in the agency) to reach an agent to inquire about insurance. If agent receives a call, agent will be busy to receive another call for a random number between 50 and 300 milliseconds.
    - Consumers can receive calls, but they are busy 80% of the time so they cannot receive calls at those times and if that's the case an agent has to try calling again at a later time
    - Each consumer has some value for the attributes: age, state, number of kids, number of cars, rent or own, household income
    - Consumers have a unique phone number
2. There are 20 agents
    - Agents can receive calls
    - Agents can make outbound calls to consumers that have called before. Outbound calls are made when a voicemail is left for the agent.
    - Each agent should be assigned to handle certain type of consumers that meet a range or value of the attributes
3. Voice mail inbox per agent exists. If a consumer calls and there are no agents available to answer because they are all busy, then calls are saved to the voicemail inbox for the best matched agent so that the agent can call back with the consumer's saved phone number.
4. There is a call router object that matches incoming calls to agents

    a. Each consumer has a set of attributes that determine how they are matched with the agents.
    b. Each agent has a set of attributes that determine they can be matched with consumers with certain attributes.
    c. If multiple agents match a consumer's attributes then an agent is picked at random
5. Create a simulation that generates 1000 consumers and 20 agents. Put random sleeps under 30 milliseconds between calls. Once all consumers have been connected with the agents, the program can end. Please output agent utilization (or 100% if all agents were utilized). It should output 3 csv files or 1 excel file with 3 tabs
    a. consumers and their attributes
    b. agents with attributes they specialize
    c. a report of how many times each consumer was called, how many voicemails were left for each agent and how many calls each agent received (this will depend on the attribute matches).