**1 Context**

We have a new project for MouldMen. They are a mould removal service with jobs across Australia. We need to schedule and allocate their jobs to minimise travel time.

**2 Summary**

We will create an program to replace the role of a MouldMen scheduler. A scheduler will receive calls from customers and schedule quote inspections (simply referred to as 'inspections'). They will then allocate the inspection to a technician. After the technician inspects the mould, they will tell the scheduler the duration of the job and how many technicians it requires. The scheduler then calls the customer again and schedules the actual job.

The scheduler may also need to schedule an equipment collection (eg. a fan) to be collected the day after the job. Sometimes the validation job will need to be scheduled to check the mould has not returned.

An inspection is a 15 minute job to determine the duration and number of technicians required to clean the mould. These will be used to schedule the clean-up job (simply referred to as ‘jobs’).

This task will detail the steps and requirements for this scheduler replacement. We will develop the program in stages (see child tasks). These stages will are designed to allow us to perform incrementally more of the scheduler’s job for a smooth transition for MouldMen.

eg. We start off Stage A. We partially deploy to MouldMen with Stage D and gradually deploy more functionality until Stage H. At Stage H the entire scheduler role will be replaced.  
Note: Stage A is not Stage 1

Here are some rough steps that we will achieve (these steps do not correspond directly to the stages):

1. Allocate simple jobs
2. Allocate simple and complex jobs
3. Allocate jobs and inspections
4. Schedule and allocate jobs and allocate inspections
5. Schedule and allocate jobs and allocate inspections, collection and validation
6. Schedule and allocate jobs, inspections, collection and validation

While we are not scheduling jobs, we will allocate the entire day in bulk. However, when we start scheduling jobs, they will come in at different time, so they will need to be scheduled one at a time. If possible, you should try to schedule the jobs in bulk.

Note the difference between ‘schedule’ (determine a start time) and ‘allocate’ (assign a technician).

**Input:**  
The inputs will change as we progress through the steps/stages. However, the inputs will typically include: job address, job start time, job duration and number of technicians required, technician availability.

**Algorithm:**  
The algorithm analyses the travel times between all jobs (e.g. inspections, jobs, equipment collection, validation, etc) and chooses the optimal route. It will also suggest when to schedule a new job to minimise unnecessary/additional travel time

**Output:**  
A schedule and allocation for inspections and jobs that allows technicians to travel with the smallest possible travel time. This will allow MouldMen to achieve more jobs with the same number of technicians.

**3 Optimisation**

We need to find the most optimal routing solution. This requires algorithms beyond a nearest neighbour algorithm or greedy algorithm. With many technicians, even small optimisations will make a large difference.

Note: The VRP is a generalised Traveling Salesman Problem.

**4 Implementation**

I have done some research and we will use Google’s OR Tools (<https://developers.google.com/optimization/routing>) to solve the Vehicle Routing Problem (VRP) with various constraints. OR Tools is a very simple Python library that is coded in C++ (very fast). It allows us to calculate the most optimal route:

OR Tools uses a time matrix that contains the travel times between all locations. Try this API for generating the time matrix: <https://mapsplatform.google.com/pricing/> (scroll down and select Routes). The matrix would be the cheapest options, but we will need to check that this is viable for commercial use.

Although it might not be necessary at our scale, this article looks interesting. It uses a neural network to guide the tree search: <https://www.frontiersin.org/articles/10.3389/fams.2023.1155356/full#:~:text=Problem%20setting-,The%20Capacitated%20Vehicle%20Routing%20Problem%20with%20Time%20Windows%20(CVRPTW)%20is,time%20window%20constraints%20are%20met.>. I do not know if we can apply this to OR Tools.

**5 Change log**

This will indicate any changes in logic or new information added to the task after it is moved to “In Progress”.

2024-02-21 Thursday: this task was restructured to be more clear and precise

Note: each stage will provide the relevant information and context. If there are any conflicts in information, assume the later stage is correct. This is because the earlier stages have simplified requirements and the later stages expand on them. These simplifications should contain a note that they will be expanded on later.