- You should be able to configure a ground station - add it to the system, so that users can just refer to them by id and have their info, like latitude, longitude, e.t.c.
    - Think of it as 2 user classes - people submitting image orders/people receiving orders (these are the ones that will be responsible for making sure the orders get scheduled)
    - The users of this system will be the satellite operators. The other users will just receive requests via an api dropbox, e.t.c. The people ordering images will not be able to go into our system to order the images. We don't need to design for that. We can just have them use an api. It is the operators that will use the system to follow the status of the orders, add satellites, remove, submit maintenance requests for satellites, outages for ground station and satellites, e.t.c.
        - In summary, image orders won't be made through our web interface. Just an api or a dropbox
- Using antenna design - what is the depth? do trait study on which antennas, e.t.c.
    - no need to go into that at all. Just take the ground station, work with their location, and have the ability to create a mask, and if it is above this level, for example, then the satellite can operate. don't have to worry about weather conditions, e.t.c.
- prince albert/gatineau height is in metres above sea level
- Will orders be collected in batches, e.t.c.?
    - Depends on the structure and methodology we chose. Ideally, each order we receive gets acted on, or alternatively, we accumulate orders, like, every 3 minutes and act on them. But the main thing is orders come in, the system takes them, processes them, and our orders are relatively agile in adapting to orders. Don't have to submit at a particular time to have the order processed. Ideally we want to minimise time that someone can submit an order and get it acted upon
- Selecting different satellites - they will have different storage capacities.
    - Each satellite will have a limited amount of images that they can store on board. They can't keep them there, they need to be downlinked.
    - if we overload a satellite, then it will be unavailable to fulfil orders when it may have been the only satellite that will be able to fulfil another future order. So now that new order will be impossible to fulfil. So we don't want to overload any one satellite
    - The only storage factor we care about is images. maintenance activities are negligible in size
- Will we have
    - have state models for the different things. The satellite doesn't know anything. You need to let the satelite know when to downlink, when to take the image, e.t.c. It doesn't know anything about that, so we need to track these statuses in our own state model
- power:
    - The document says we assume the sun has sufficient power to sustain. However, satellites will enter eclipses. We will need something in our system to let us know if the satellite is in eclipse or not. If it is in an eclipse, we need something to sort of ration the power. Say eclipse takes 20 minutes, so within this period, the satellite will be able to take only 6 images, so we may need to reject and adjust our schedule so that the satellite battery will not get too low from it taking too many images for example
    - our system should be able to tell eclipse periods (python library exists, feed TLE, feed time, then it tells you am i in eclipse or not)
        - sun is blocked by the earth
        - working on batteries
- 2 possible outages:

- spacecraft
- ground station:
    - say gs gatineau, outage for 2 days. any contact that you have to update that satelite, or downlink images, will no longer be able to be carried out. Our system will need to reschedule these things. Move those uplinks to a different ground station, move downlinks to different ground station as well
    - System will need to have a capacity to react to such outages. You will have to move those activities to other resources, as well as not schedule additional resources using those resources
- system can communicate to ground station to send maintenance request
    - maintenance requests may be viewed as an outage. maintenance may prevent other activities from taking place, it may not. The difference between maintenance and outage is that we are able to schedule it to a convenient time

4 things to consider in schedule (at least we can think of as of now):
- storage capacity
- outages
- maintenance activities
- eclipses

- image resolutions are static, configurable parameters
    - if a new satellite gets introduced, it may have different kinds of parameters, and those may need to be tweaked.
    - there will be upper limit for these
- request priority
    - will be provided by operator with the request - 3 levels (high, medium, low)

- for repeat cycle
    - number of times it repeats, and how often it repeats. The difference between those two is this:
        - e.g. flushing the ram on-board the satellite
            - you want that to occur every 2 days. you want it to occur 14 times. So you have 2 things - every 2 days, and it will happen 14 times

Sample orders will be sent to us middle-end of next week with concrete numbers

on document, it says that the schedules should be automated, but also user configurable
Does that mean we need to keep track of every schedule sent to the ground station so that that schedule can be reconfigured?
- kinda - maybe an operator wants a schedule to be created and wait for approval before it is sent. You have user intervention - you have steps within the workflow where the user can see what's going on, and can push things forward, e.t.c. so they want to be able to control that.