Potential Non-Functional Requirements

- Minimal number of clicks
- Non-repetition of tasks
- Code free experience
- Objects should work seamless across different types of Unity environments (non-quadrilateral spaces)

Number of clicks

- The number of clicks rule isn't necessarily correct as shown by studies rather it should be relative to the application
- If you can construct a user interface that's easy and pleasurable to use with a lower click count than the competition then it's a success
- Count the number of clicks it takes to perform tasks in VRTK

GUI Design

- Look into different types of UI design and layout
- The F-Shaped Pattern to enable content skimming
- Important information on the LHS of the UI
- Stuff like that

Functional Requirements

- This will be the set of features required for the creation of working product
- Additionally it will also require the product be live on the Unity Asset store
- Deemed successful if it implements the required features and is live on the store

Potential Feature Idea List
Controllers

- *Controllers present with object interaction*
- *Job Simulator hands disappear method with object interaction*
- Object interaction snaps to a fixed on the object that is picked up
- With objects you have to setup the scripts for both hands independently: setup the default as both where you slot one script in and allow the user to specify they want the hands to different things and so just open up two controller sub-boxes
- *Button mapping*
- *Ghost or Physical controllers in-game option*
    - *https://youtu.be/Sp0knS9S8fo?t=250*

Display: User's hands

- *Change model to different model than standard controller*
- *Change model of spectator view but keep in game view standard controller*

Display: User's head

- *Similar to changing the model of the user's hands for the spectators they also changed the user's head model*

Decoupling & Repetition

- *One problem that accidentally deleting one object causes a lot of problems that the user might not be aware about and so how do we make the user aware of such problems (possibly not allow them to not do certain things)*
- *Explain more through the GUI but keep it simple to use*
- *Another problem is you are setting up the same thing over and over again. So why not make the exception if you want it to do something different.*
    - *E.g. To make a "pick up" object drag and drop one script which preloads as much as possible (preloads that both hands will pick up the script upon interaction with the object)*

Interactions:

- *What happens when you move a controller over an intractable object does the controller vibrate, highlight the object, etc.*

Interaction: Drawers

- *Universal drawer than can be used in non-quadrilateral spaces*
- *Drag and drop script and you specify the front of the drawer and a second point to indicate how far you are able to pull it (default to front if second point not given)*
- *Specify the axis on which the drawer moves (default to 90)*
- *Get this working on a box and then figure out how to put stuff in it later*
- *Example Scene: Open the drawer to find a torch which you can turn on / Open the drawer to find a keycard which opens a door (Star Trek sliding door complete with whoosh)*
    - *Maybe attach to front panel of drawer which is a separate object from rest of drawer and specify the*
    - *Work independent of the Unity global axis / allow for circular, rotated spaces*
- *Possibly the ability to toggle gravity on and off for a drawer. That is a VRTK drawer just sits in midair if there's no cabinet surrounding it. Actually this might be troublesome (maybe do this)*

Interaction: Cupboards

- *Variant of drawer*
- *Tackle after drawer*

Interaction: Levers

- *Pull object on an axis*

Interaction: Doors

- *Push object on an axis*

Interaction: Buttons

- *Different classes of button*
- *Buttons on objects and reseting their global axis to be that of the object*
- *Anchoring buttons in general onto objects and surfaces*
- *Timing associated with events and buttons*

- *Spamming the button being pressed*

Movement

- *Configuration of where the user is able to teleport*
- *Different types of display and arcs*
- *Teleport onto object (“floor tag” tag system)*
- *Teleport to specific location*

State Machine

- *Predeveloped and possible inclusion*

Loading Between Scenes

- *Single drag and drop with a selection of parameters to change the skybox, etc.*

Automated Testing

- *Capture movements and playback force ghost style*
- *TrialsHD Method of capturing the inputs and recreating them using the captured inputs*
- *Would it be possible to record the motions and then play them back without the HMD attached to the device*