

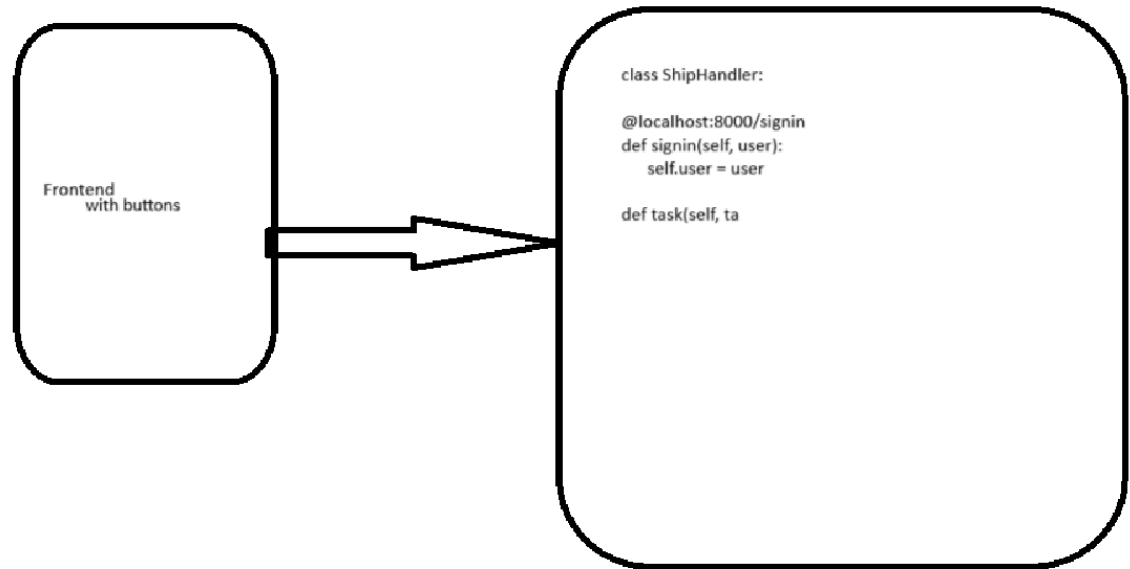
Design Doc

Frontend:

- React or Python

Backend:

- Python
- Tkinter
- Connect frontend to backend with flask
- (Melanie) User sign in - Melanie
 - Just sign in using name
 - Have a variable for user name
 - Show name on home page
- Choose task between Balancing and Loading/Unloading - Melanie
 - Once task is chosen, offer popup to upload manifest file
- (Melanie) Read and parse the manifest (Get a string)
 - Read slides on manifest format, probably need to parse it line by line
 - Handle NaN Unused, and create matrix and visualize the state of the ship
 - https://www.dropbox.com/scl/fo/rugw8713icmaqy9t587eb/AD_rHdevYqFEIM4jAt9AYRc?rlkey=lllt6iibn4740i7yyw997hv3m&dl=0
- Visualize the state of the ship
- Balancing:
 - Warning at top if ship is unbalanced, and button to run balancing algorithm
 - Give steps
 - Manifest updated and written with OUTBOUND
- Loading/Unloading:
 - Offer two buttons, one for load and one for unload
 - Manifest updated and written with OUTBOUND
 - Unload:
 - A grid visualization of the ship manifest
 - User will select which one to unload
 - Hovering over a box will show details
 - Load:
 - Ask for details like Name of the container, additional information
 - Then run algorithm
- Update logs
 - Update after every move
- Last: Work on animation of algorithm



Functionality:

- User will pick what task to do



- Will load or unload containers
- To unload will click on container which will be added to list
- To load, will be prompted by software for name and weight, which will be used to

update the manifest

- Sign in box with login will be present on every page
 - This is will also update the manifest with the new login event