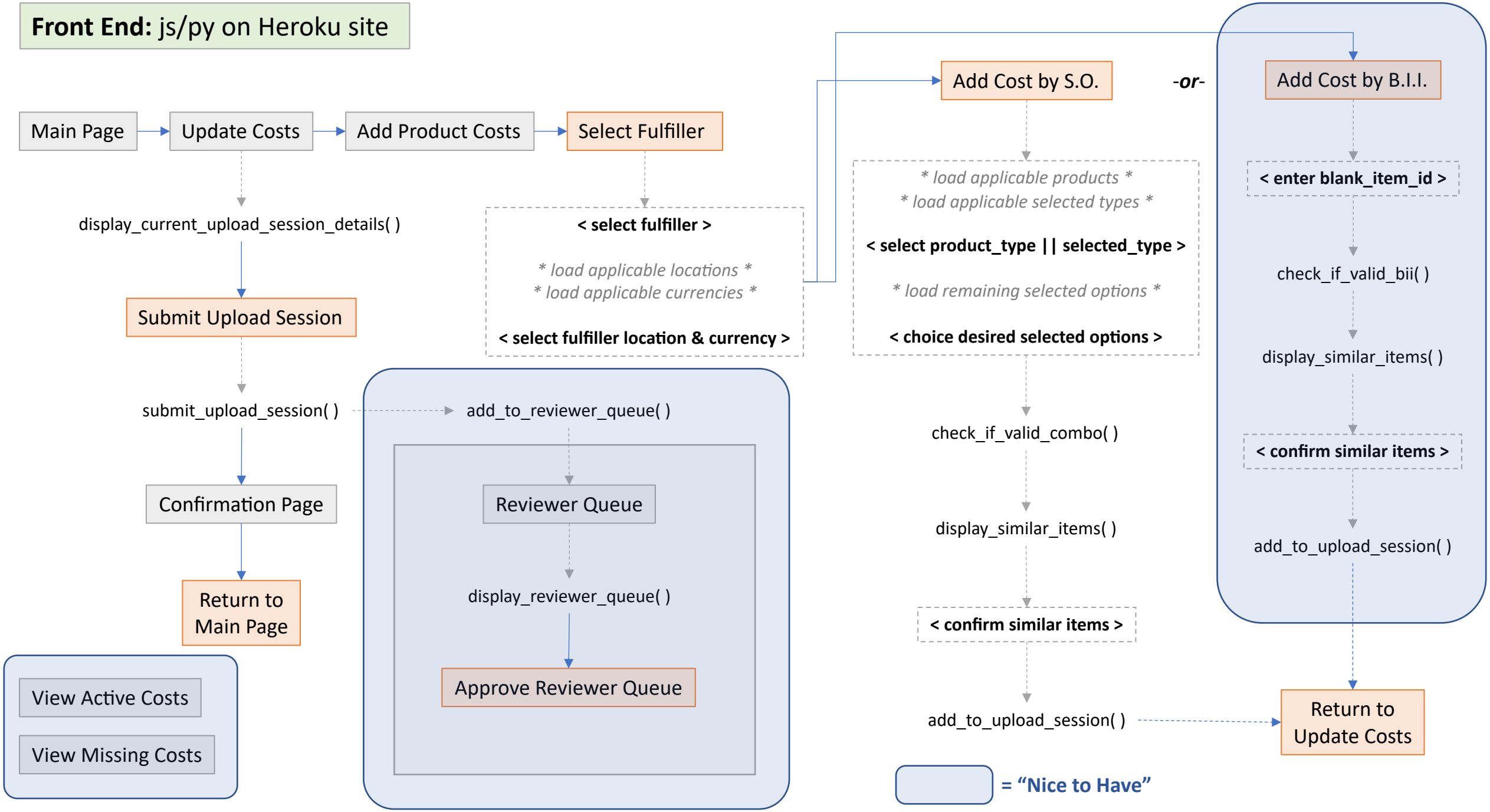
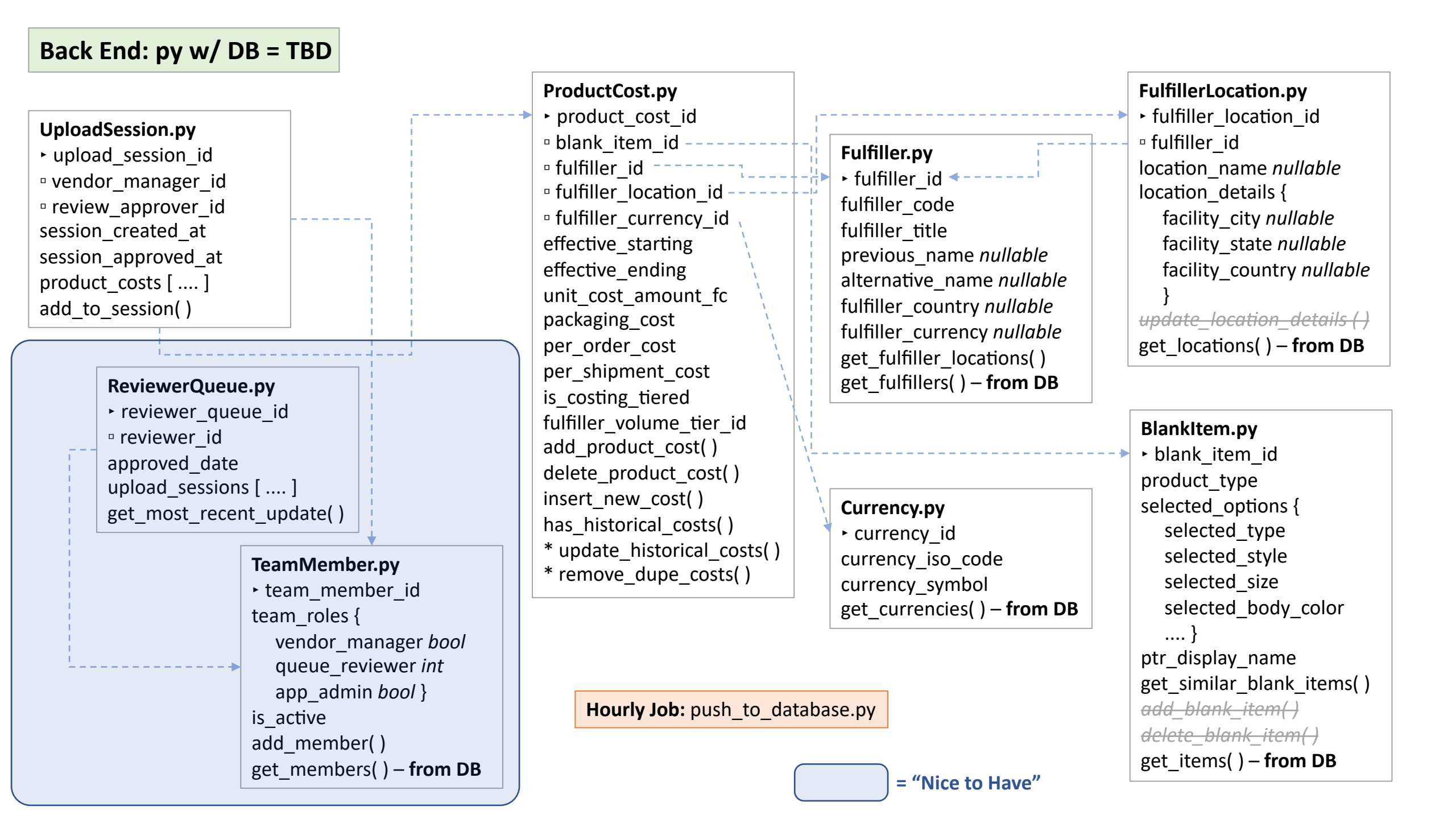
Costing Project- Phase 1

Objective: MVP of standalone webapp

- accept cost data
- error validate & process
- update dates & store in DB


Front End: js/py on Heroku site
Main Page
Update Costs
Add Product Costs
Select Fulfiller
display_current_upload_session_details( )
Submit Upload Session
submit_upload_session( )
Confirmation Page
Return to Main Page
< select fulfiller >
* load applicable locations *
* load applicable currencies *
< select fulfiller location & currency >
Add Cost by S.O.
-or-
Add Cost by B.I.I.
* load applicable products *
* load applicable selected types *
< select product_type || selected_type >
* load remaining selected options *
< choice desired selected options >
check_if_valid_combo( )
display_similar_items( )
< confirm similar items >
add_to_upload_session( )
< enter blank_item_id >
check_if_valid_bii( )
display_similar_items( )
< confirm similar items >
add_to_upload_session( )
Return to Update Costs
add_to_reviewer_queue( )
Reviewer Queue
display_reviewer_queue( )
Approve Reviewer Queue
View Active Costs
View Missing Costs
= "Nice to Have"


Back End: py w/ DB = TBD
UploadSession.py
▸ upload_session_id
▫ vendor_manager_id
▫ review_approver_id
session_created_at
session_approved_at
product_costs [....]
add_to_session( )
ReviewerQueue.py
▸ reviewer_queue_id
▫ reviewer_id
approved_date
upload_sessions [....]
get_most_recent_update( )
TeamMember.py
▸ team_member_id
team_roles {
vendor_manager bool
queue_reviewer int
app_admin bool }
is_active
add_member( )
get_members( ) – from DB
ProductCost.py
▸ product_cost_id
▫ blank_item_id
▫ fulfiller_id
▫ fulfiller_location_id
▫ fulfiller_currency_id
effective_starting
effective_ending
unit_cost_amount_fc
packaging_cost
per_order_cost
per_shipment_cost
is_costing_tiered
fulfiller_volume_tier_id
add_product_cost( )
delete_product_cost( )
insert_new_cost( )
has_historical_costs( )
* update_historical_costs( )
* remove_dupe_costs( )
Fulfiller.py
▸ fulfiller_id
fulfiller_code
fulfiller_title
previous_name nullable
alternative_name nullable
fulfiller_country nullable
fulfiller_currency nullable
get_fulfiller_locations( )
get_fulfillers( ) – from DB
Currency.py
▸ currency_id
currency_iso_code
currency_symbol
get_currencies( ) – from DB
FulfillerLocation.py
▸ fulfiller_location_id
▫ fulfiller_id
location_name nullable
location_details {
facility_city nullable
facility_state nullable
facility_country nullable
}
update_location_details ( )
get_locations( ) – from DB
BlankItem.py
▸ blank_item_id
product_type
selected_options {
selected_type
selected_style
selected_size
selected_body_color
.... }
ptr_display_name
get_similar_blank_items( )
add_blank_item( )
delete_blank_item( )
get_items( ) – from DB
Hourly Job: push_to_database.py
= “Nice to Have”