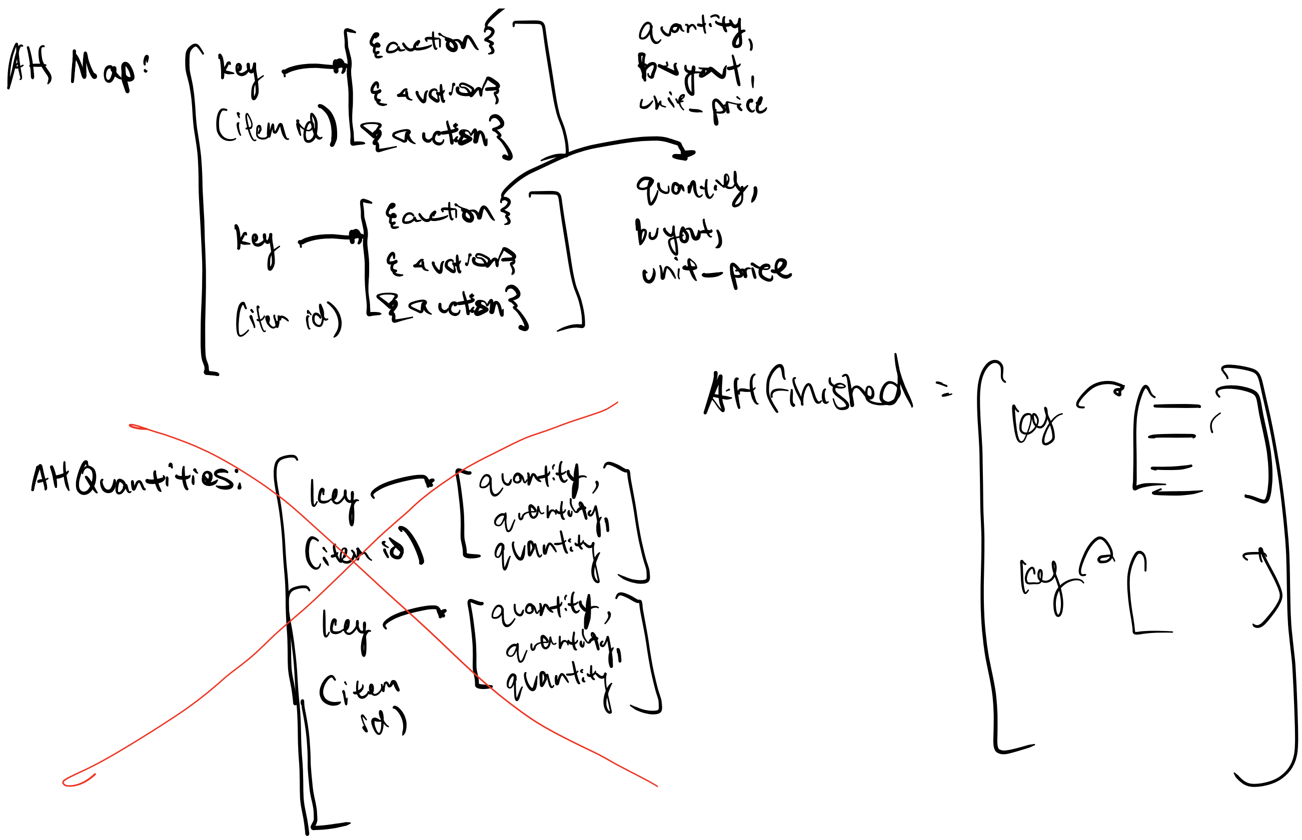


Avg: for each item:
sum up quantities
reduce AH Map to
avg. by prevValue + currentValue.quantity * currentValue.buyout ||
currentValue.unit-price
total
/quantity.

For the app to be useful:
- notifications on price drops X
- market history
  item: { name: "____"
  history: [________] }
  ↓
  low price? → avg of lowest 10% of Quantity?
  (lowest price w/ quantity.)
  Avg Price!
- Grab recipes & check ROI on Labor.
- use market history to compare today's prices with the last week
- Graph data.

```
Data in: { —
    auctions: [ { id: ______
                  item: {
                        id: ______
                        . . .
                  }
                  quantity:
                  buyout:
                  (unit_price):
              ]
}

Convert
   AHMap : [ key → { item id:
                     item name: null      ↱ buyout , quantity
                     item Auctions: [ {    }, {  } ]
                     quantity:
           ]
```

```
findAverages = (AHMap) => {
  let newMap = new Map(AHMap)   //? Should it be ... AHMap ?
    [... awMap.keys()]. forEach ((key) => newMap.set (key, {... newMap.get(key), avg:
    newMap.get(key). itemAuctions . reduce( (avg, value, _ , { length) => avg + value/length, 0));
}
```

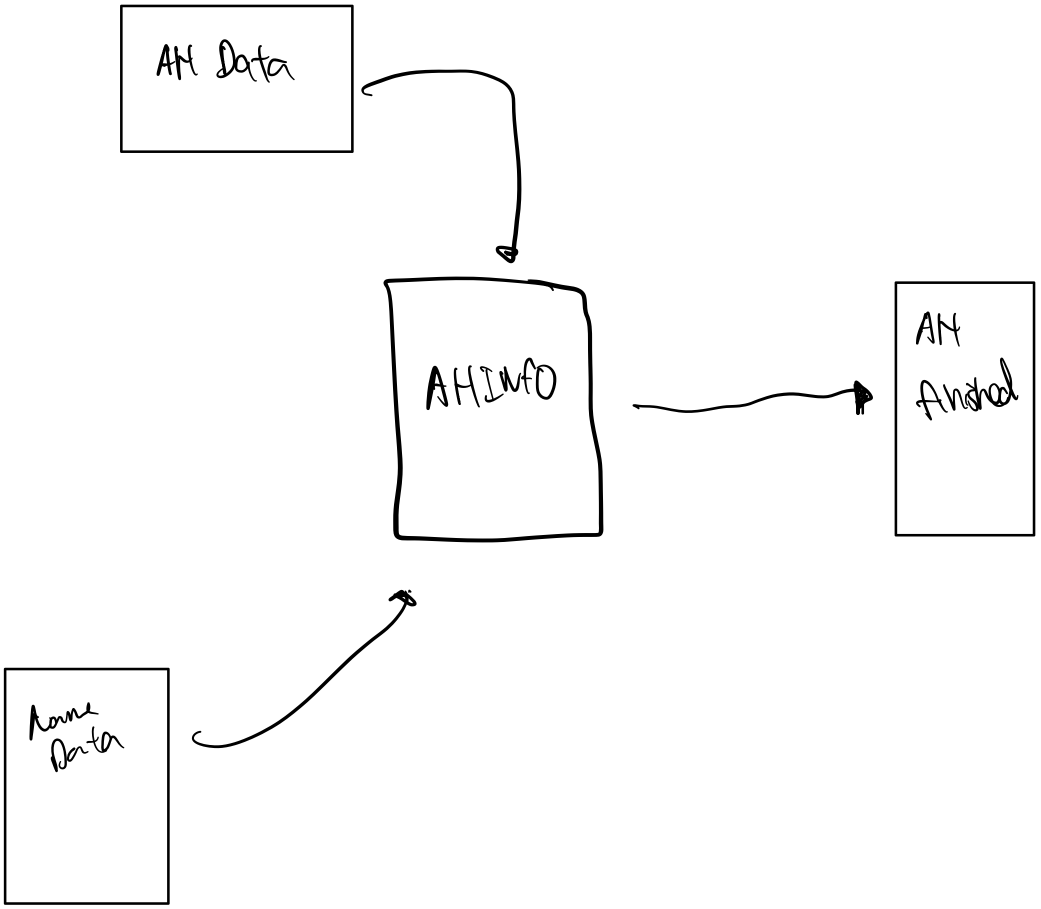
~~findAverages = (AHMap) => {~~

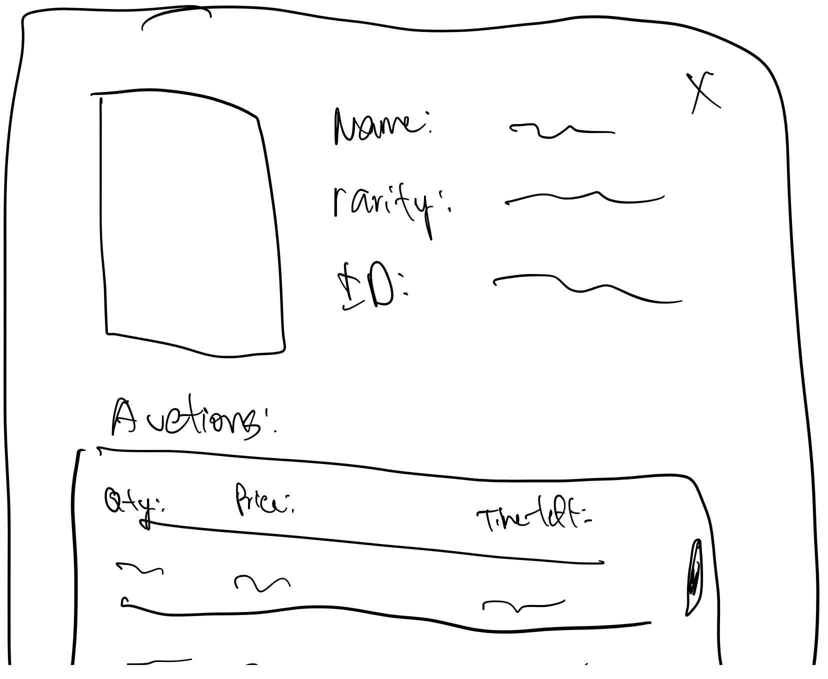
~~let avgs = [...AHMap.keys()]. forEach((key) =>^ avgs.set(key, AHMap.get(key). reduce((avg, value, _, {~~

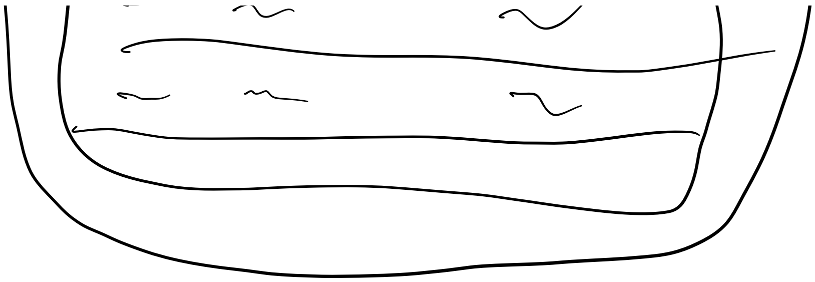
~~=> avg + value.quantity * value.buyout / AH quantities.get(key));~~

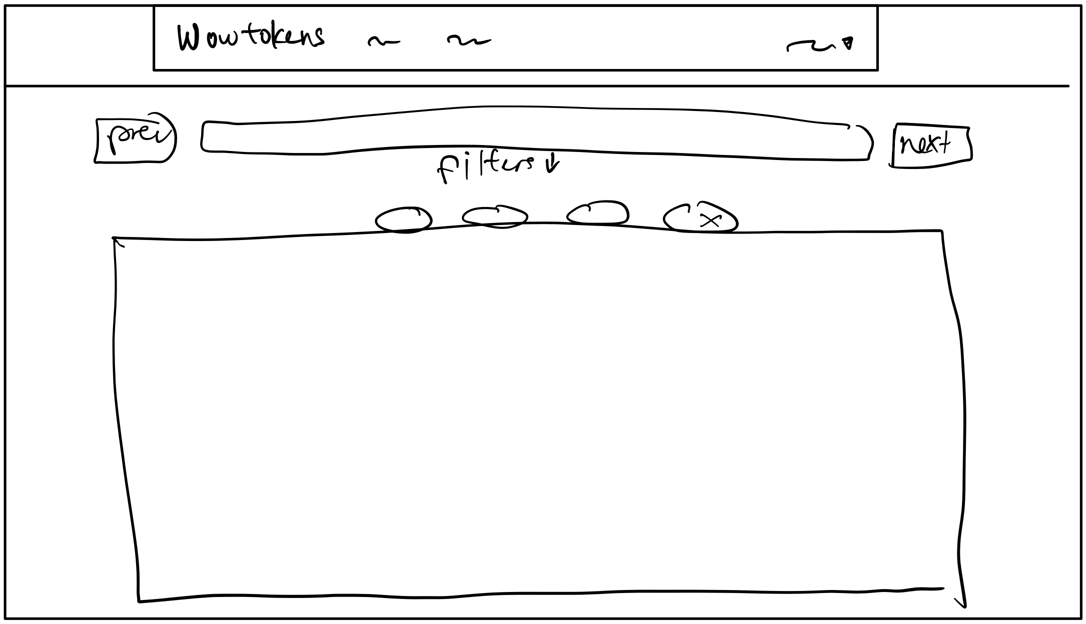
~~set Averages (avg);~~

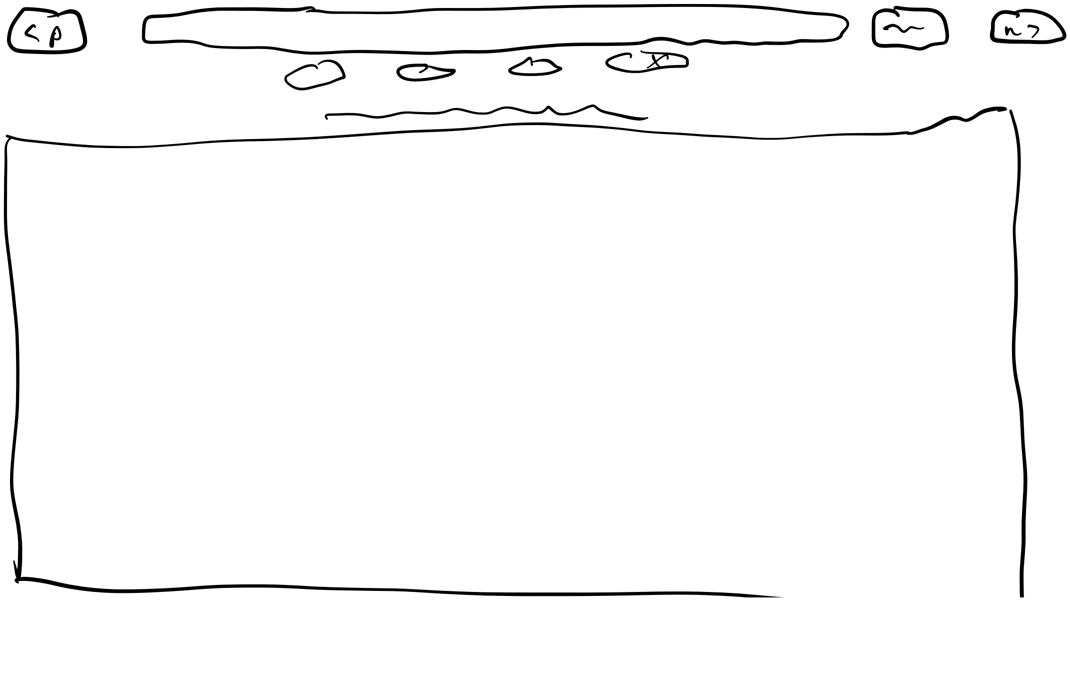
~~}~~


AH Data
AHInfo
AH Anshed
Name Data

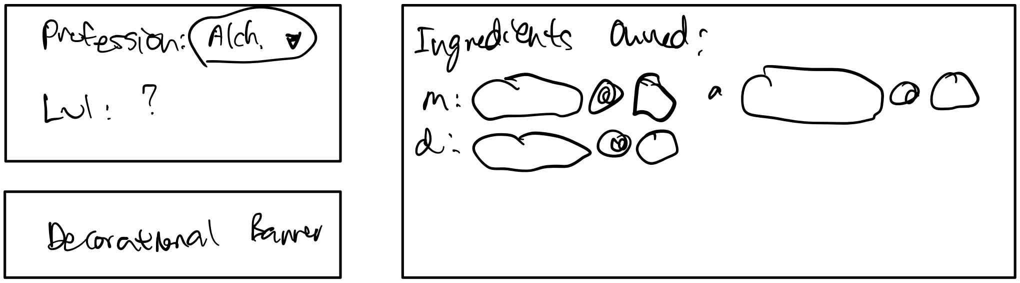

X
Name:
rarity:
ID:
Auctions:
Qty:
Price:
Time left:




Wowtokens
prev
next
filters ↓

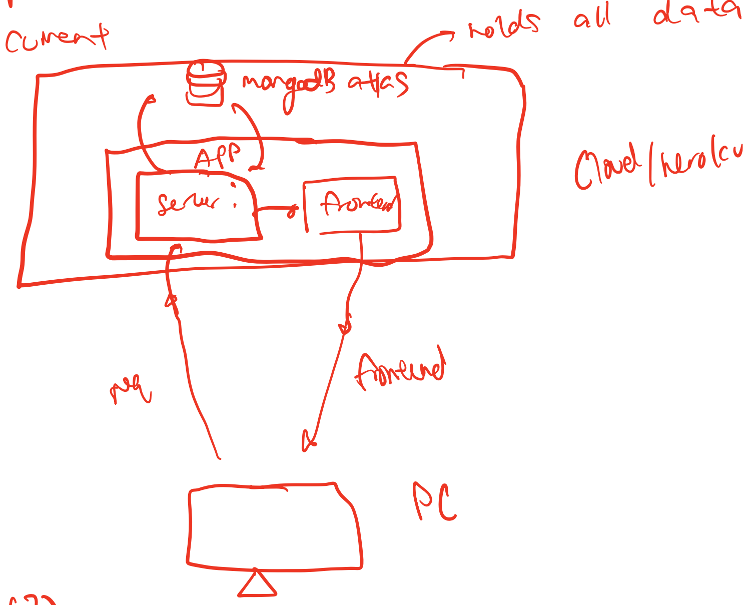

< p
n >

CALCULATOR SCREEN

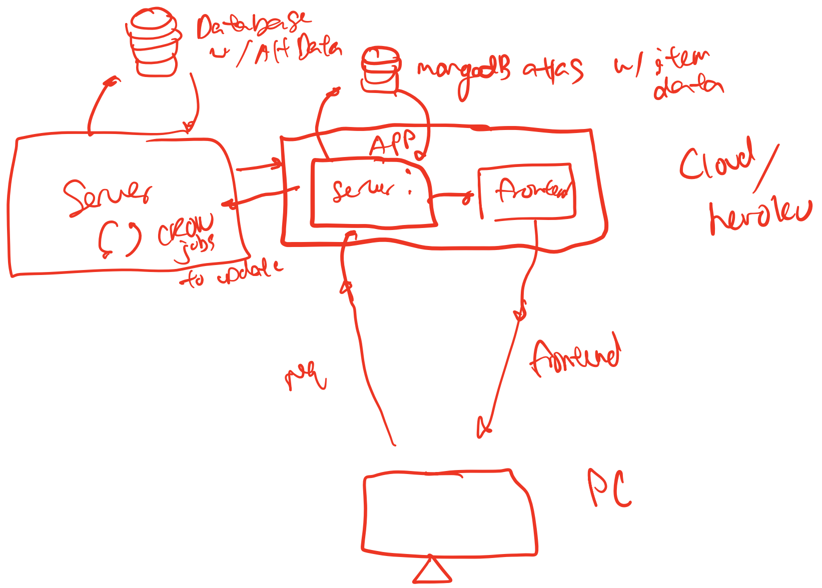


Recipe	Cost	market price	Profit	Profit w/ owned mats

App Architecture switch(?)



NEW (?)


Database
w/ AH Data
mongodB atlas w/ item data
Server
( ) CRON jobs to update
APP
Server :
Frontend
Cloud/
herolcu
req
Frontend
PC