

# Development Logistics Math (Thoughts)

ASPIRATION: **App should do what takes 8 hours, in 8 clicks.**

- 2 months = 63 days = 9 weeks
- 9 weeks = 8 x ({5 days of coding & testing} + {1 day of review}) + 1 week of launch-checks
- Our aim should be 6 core functionalities; that's a deliverable functionality, every 10 days.

## Functionality List

#	Name	Description
1.	Portal	Sign-up portal which securely captures and stores all relevant user information in a database
2.	Reporter	Report every 10 minutes if a listed product is <b>[IN STOCK]</b> or <b>[OUT OF STOCK]</b> (with time-stamps)
3.	Repricer	Based on defined pricing rules, <b>AUTOMATICALLY</b> update price if Amazon Price is less than Walmart Price for listed product (send notification)
4.	Scraper	Scrape UPS, FedEx and USPS to find tracking numbers that are based off of the exact postal code and city of another parcel and correctly replace it with the TBA tracking numbers
5.	Tracker	Display breakdown of <b>ORDERS, TRACKING NUMBERS</b> , as well as <b>SALES, PROFITS and EXPENSES</b> . Send monthly email for tax purposes.
6.	Chatbot	Install chatbot (like Tidio) for automated customer support

## Timetable

September	October	November	FINAL
-----------	---------	----------	-------

WEEK	M	T	W	T	F	S	S	DELIVERABLES
1	20	21	22	23	24	25	26	
2	27	28	29	30	1	2	3	Portal
3	4	5	6	7	8	9	10	
4	11	12	13	14	15	16	17	Reporter
5	18	19	20	21	22	23	24	Repricer
6	25	26	27	28	29	30	31	
7	1	2	3	4	5	6	7	Scraper
8	8	9	10	11	12	13	14	
9	15	16	17	18	19	20	21	Tracker & Chatbot

## ANAS R CONCERNS

One major concern we've identified is that **Amazon doesn't have a native API to 'purchase items'**. Therefore that's going to be a **TO BE DETERMINED** scope item (we need to research and see how we can do that with it being stable)

Any non stable items are usually a pain to maintain as, if anything changes, it just doesn't work without constant tweaking of code

### 3 Ways Things Will Work:

1. API (most stable, widely used and easy to support).  
We will try to do everything through native or third party API's
2. Data Scraping (will be used in various places API is not available).  
This is again a little bit problematic with support if any pages or UIs change on Amazon side)
3. Data input manual (same as #2 except now we are not pulling data we are reading the pages of Amazon or another website and then inputting the userid/pwd + order details all manually)

Will automate manual data entry by scanning pages and specifying input areas and automatically entering them etc (this is worst case incase we can't find an Amazon orderer api) from a third party service

Asked Nayan for updates on the order submission on Amazon? We need to figure out if that's possible.

Current flow - customer buys from clients Walmart account, customer manually orders the same product from their own personal Amazon account (from the suppliers store)

## GOOD STARTING POINT

- We have our own software (which has been made),
- which has user logins,
  - we already have the portals where people can register and
  - use the login for the trials and stuff like that.

### Observations:

- Valiant is a Done-For-You (DFY) Automated Online Store
- Valiant's Amazon drop shipping service works in a three-step process.

**Step 1.** The client sets up their store using Valiant Consultants' over-the-shoulder training videos.

**Step 2.** **The client provides the Valiant Consultants team access to the store they have created.**

**Step 3.** Valiant leverages its skilled and experienced staff to help grow the client's store and take sales to the next level.

- With an app our ONLY concern is **FUNCTIONALITY**.  
We don't have to worry about leads or conversions. We can code, test and have the client play with the software while we keep developing it.

**TARGET:** USER who buys products from Amazon and sells on Walmart for a profit.

## HOW SHOULD THE APP BEHAVE

The USER should be able to log in to THE APP

The USER should be able to immediately see their ORDERS, SALES, PROFITS and EXPENSES

The USER should be able to immediately see their listed products and what's in [\[IN STOCK\]](#)

When the price changes, THE APP should automatically change prices (As Per Rules) and notify USER

THE APP should automatically scrape tracking numbers to replace with TBA tracking codes

If the USER requires customer support, an automated chatbot should be available.

The APP should be able to handle multiple USERS and heavy loads. Available at all times with disaster-recovery.

## THINGS TO DO & RESOURCES NEEDED

- a. We're going to need lots of testing data - WALMART TEST ACCOUNT, PRODUCTS. SAME FROM AMAZON.
- b. We have to be able to create unlimited dummy user stores
- c. We'll need to be able to list products in bulk
- d. Users should be able to enter credit card details securely
- e. We'll need to be able to access a copy of this Google Sheet of products (maybe)
- f. We need Walmart and Amazon API keys, as well as other 3rd Party APIs
- g. We need to conduct trials on OA Genius or EComCircles to see what Steven does.  
Us creating an account is not enough, as we don't know how Valiant uses those software apps.
- h. We need to figure out how to **scrape UPS, FedEx and USPS to generate tracking numbers**
- i. Then we need to automate TBA code replacement
- j. Imagine a page with a button called "SCRAPE". What else is on the screen? Talk me through how the user got here and if successful, what should they see.

## TASK LIST

Statement: TEAM goes on to Amazon and finds examples where it's cheaper on Amazon and more expensive on Walmart.

CURRENT WORK REQUIRED:

1. Search for a product on Amazon (Product AA)
2. Get product's price (AA.price)
3. Find same product on Walmart Marketplace (Product AW)
4. Compare: If (AA.price < AW.price), then send Product AA details to Google Sheet

HOW SHOULD THE SOFTWARE IDEALLY BEHAVE:

Establish PULSE which periodically checks both Amazon and Walmart for products.

When products match, **ONLY send product details when AMAZON PRICE IS LOWER THAN WALMART.**

---