The engineering spec for the MVP through V1, including rough estimations on work required in engineering days.

[Product MVP PRD](https://docs.google.com/document/d/1-kQjq86UdMJFS-Agw-xHnJmr9O-ONTetp4I149xiJ-k/edit)

## Frontend

React + redux

## Backend

**V0.5:**

2: Set up plumbing on Heroku (Linux + Python + MYSQL, NGINX if easy if not Apache) and create new Github private project

0.5: Create a rough data-flow and architecture diagram

0.5: Spec RESTFUL API

0.5: Define and create sql tables (publisher, store, sku, store\_sku)

0.5: Scaffolding for Python project and create YAML configuration library

2: CRUD endpoints for skus, Publishers, and Stores, and SKUS Batch GET

0.5: Create url schema for stores and individual skus

1.5: Integration with third party payment processor

8 engineering days

**V1:**

0.5: Paginated skus

1: Searchable skus, either basic mysql fuzzy match or smarter elasticsearch or equivalent (+1 day)

1: Searchable stores

0.5: Create consumer table

0.5: CRUD endpoints for consumers, signup and create account, login validation

0.5: Create preferences table

0.5: CRUD endpoints for updating preferences

1: Create url schema for store’s with batched skus, and ability to generate link with multiple skus

2: Generate analytics breakdown for Publishers

1: Purchase and create an account at the same time

1: Login and get payment and user information from a new endpoint

1: Send a confirmation email upon purchasing an item (place behind a Celery Queue)

0.5: Set up Caching via Redis

0.5: Cache behind Cloudflare

1: Documentation and Graphite integration

1: Sentry alerts

13.5 engineering days

## **RESTFUL API**

|  |  |  |
| --- | --- | --- |
| GET | /publishers/ | Get all publishers |
| GET | /publishers/:publisher | Get one publisher |
| POST | /publishers/ | Create a publisher |
| PUT | /publishers/:publisher | Update a publisher |
| DELETE | /publishers/:publisher | Soft Delete a publisher |
|  |  |  |
| GET | /stores/ | Get all stores |
| GET | /stores/:store | Get one store |
| POST | /stores/ | Create a store |
| PUT | /stores/:store | Update a store |
| DELETE | /stores/:store | Soft Delete a store |
| GET | /stores/:store/skus/ | Get all skus for a store |
| GET | /stores/:store/skus/:sku | Get a sku for a specific store |
| POST | /stores/:store/skus/ | Create skus for a store |
| PUT | /stores/:store/skus/:sku | Update a sku in a store |
| DELETE | /stores/:store/skus/:sku | Soft Delete a sku from a store |
|  |  |  |
| GET | /skus/ | Get all skus |
| GET | /skus/:sku | Get one sku |
| POST | /skus/ | Create a new sku |
| PUT | /skus/:sku | Update a sku |
| DELETE | /skus/:sku | Soft Delete a sku |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

## Database schemas

**Resources needed:**

Markets (App stores created by publishers)

Products (product templates)

* uuid
* name
* description
* price
* s3\_url (image of item)
* rel - category
* rel - sub category
* rel - options

Orders

Publishers (App owners)

Supplier (Where the products come from)

* uuid
* user uuid

Users

* first name
* last name
* email
* contact number
* password salt
* password hash

Categories

Sub Categories