1. Init

Каталог продуктов с категориями и под-категориями
Только API на Django Rest Framework, морда пока что на Django REST Swagger

Models:

- Category
- Item

Endpoints:

- **CategoryListView** Список Категорий с фильтром по категории (только указанная категория с её под-категориями)
- **ItemListView** Список Товаров с фильтром по категории, сортировкой по цене
- **ItemDetailView** Детали Товара
- **ItemAddView** Добавление товара
- **CategoryAddView** Добавление категории
- **AuthView** Авторизация

http://www.django-rest-framework.org/
http://django-rest-swagger.readthedocs.org/en/latest/

CategoryListView

- id
- name
- sub_category

ItemListView

- id
- name
- price
- image_url
- url - полный URL на DetailView для элемента
- category_name

ItemDetailView

- id
- name
- price
- category_id
- category_name
- description
- image_url

ItemAddView

- name
- price
- category_id
- description
- image_url

CategoryAddView

- parent
- name

AuthView

- login
- pass

Обязательно настроить админку, что бы можно было добавлять все от туда перед тем как начать делать FE

2. FE

- AngularJS
- Выводим список продуктов и список категорий. Для продуктов нужно сделать DetailView, где будут все поля.

3. CacheBack

Перед началом этого раздела создать ветку и работать в ней. Для ветки нужно сделать Pull Request

ItemDetailView и **CategoryListView** через кэш, инвалидация на сигналах. Django Cacheback работает с Celery.

Пример инвалидации - http://codeshare.io/PI8Q9
https://django-cacheback.readthedocs.org/en/latest/

Далее все новые запросы должны оборачиваться в кеш.
Вообще использование кеша на листах проблематично, но попробовать стоит.

4. Авторизация и сортировка

Список товаров можно сортировать по цене

5. Rates

Возможность оценивать товары
Для каждого товара возвращать среднюю оценку в API

6. Comments

Возможность добавлять комментарии к товарам
На списке показано количество комментариев у товара

7. Search powered by Haystack

Все ListView переделывает с использование Haystack. API остается тот же, т.е. Front-End не должен понять что что-то изменисоль.

В качестве backend - можно Whoosh, но со след пунктом будут проблему, так что лучше сразу ElasticSearch
ElasticSearch http://haystacksearch.org/

8. Faceting

http://django-haystack.readthedocs.org/en/latest/faceting.html
Для категорий показывать количество результатов.
Например:

Filter by category:

- Book (150)
- Magazine (15)
- Paper (20)

9. Магазины

- Акаунты-магазины могут добавлять товары
- Для каждого магазина можно установить %, который накидывается на цену выставленную магазинов на товар
- Страница со списком магазинов и страница магазина только с их товарами

- У товара есть количество в наличии
- Если 0 - нет в наличии, меньше 10 (можно настраивать количество) - “заканчивается”

10. Процесс покупки

- Товар можно добавить в корзину
- Можно просматривать корзину
- Можно купить (всю корзину сразу) (процесс оплаты имитировать)

11. Акции

- Для любого товара можно создать акцию.

- Акции могут создавать аккаунты-магазины и только для добавленных ими товаров
- Акция: описание, новая ценна, период акции
- В списке товаров и на странице товара старая ценна перечеркнута и показана новая
- Покупка по акционной цене