Суть идеи заключается в создании и поддержке интернет сервиса, который позволяет формировать личную и всеобщую базу данных, включающую в себя информацию о Товарах (включая все, что можно купить или на что можно потратить деньги), месте их продажи (магазин, или любое другое место продажи Товаров). На основе базы данных организуется поисковая система, которая позволяет пользователю получать необходимую информацию из базы данных, применяя любые запросы, предлагаемые сервисом. Сервис отвечает на 4 основные вопроса: что, где, когда, почем.

Сервис (что где когда почем) позволяет:

1. Накапливать личную информацию о расходах. Это дает возможность анализировать свои расходы.
2. Сравнивать цены на товары в разных магазинах и экономнее планировать свои расходы.
3. Находясь в магазине получать информацию о ценах на аналогичные товары в других магазинах и не делать опрометчивых покупок.
4. Изучать ассортимент магазинов заранее и не тратить время на поездки без покупок.
5. Проводить маркетинговые исследования.
6. Нужно реализовать механизм учета НДС.
7. Дата по умолчанию сегодня
8. Поля добавились: колво, сумма, НДС (вкл/выкл)
9. Сортировать по дате.
10. Необходимо реализовать механизм анализа похожих слов (рус/укр/engl).
11. Необходим механизм отслеживания пользователей-врунов цен и отодвигание их назад по поиску.
12. Добавилось поле Совпадений. Хранятся не все строки пользователя а только ключ позиции и пользователь. Возможно есть 2 базы данных. Одна локальнее к пользователю. 2-ая не хранит информацию о пользователе а только о товаре количество подтверждений.
13. Можно клацнуть на цене сущ. Позиции и она сдублируется.
14. Пользователи оцениваются по индексу доверия. (он учитывает стаж пользователя правдивость на протяжении стажа).
15. Сортируется отбор по индексу доверия.
16. Прежде чем запускать сервис необходимо его патентовать.
17. Собранная мировая статистика может быть полезна.
18. Возможно необходимо учитывать не только день но и «время» цены.
19. Необходимо разработать текст для патента.
20. Когда ставишь +1 на строке в отобранных строках позиция попадает к пользователю в список. Это нужно когда набираешь свой чек.
21. Тогда получается, что в поисковых строках не нужно количество и сумма. Эти поля есть только в базе пользователя.
22. К совпадений рассчитывается количеством пользователей, подтвердивших.
23. Пользователь может просто подтвердить цену, или же внести ее в покупку.
24. Добавить фильтр последняя цена.

Объекты (классы):

1. Пользователь
   1. Login
   2. Password

Методы:

1. Установить новую цену на товар на дату в магазине.
2. Добавить магазин
3. Занести магазин в избранное
4. Занести товар в избранное
5. Указать цену ???
6. Подтвердить цену ???
7. Занести покупку
8. Завести чек

Избранные магазины

* + 1. Искать товар (выдает запросы)
    2. Подтвердить цену
    3. Сделать покупку

1. Товар
   1. Наименование
   2. Производитель
   3. Расфасовка
   4. Штрихкод
2. Покупка
   1. Товар
   2. Дата
   3. Цена
   4. Количество
   5. Сумма
      1. Создать покупку
      2. Редактировать покупку
      3. Сохранить покупку в XML
      4. Сохранить покупку на сервер
3. Магазин
   1. Название
   2. Адрес
   3. Координата GPS
      1. Добавить новый магазин
      2. Редактировать магазин
      3. Сохранить магазин в базу данных
4. Чек
   1. Номер
   2. Покупка 1
   3. Покупка 2
   4. Покупка 3
5. Окно программы
   1. Элементы

Нажатия на элементы окна вызывают Public методы классов Магазин, Пользователь, Чек, Товар, Покупка.

1. База данных на сервере

Получает запросы от пользователей

1. XML-файл у пользователя на устройстве
2. Администратор базы данных
3. Регион
4. Охват

Логгирование

Нарисовать HTML дивами. Блочный принцип.

Смотри stackoverflow

Разобраться с сессиями и +куками

+Как устроены связи в mySQL, писать одновременно в несколько таблиц, сложные запросы.

Google maps API наносить магазины на карты, определять набор магазинов по радиусу и по географическим по единицам (страна, область, населенный пункт, район и т.д.)

Пользователь может логиниться, а может и не логиниться. Если не логинится, то мы можем писать ему куки и потом отслеживать. Если логинится, то вся информация о нем хранится в БД.

При появлении нового коннекта к нашему сайту, мы просматриваем куки и если клиент еще не был у нас, то создаем его аккаунт. (записываем ему куки, делаем запись в БД) Если такой user запоминает свои товары, то мы сохраняем их в БД с его ID. Далее, когда он подключается повторно, мы вычитываем его Товары и ищем в первую очередь в его записях.

Вирус на JS ходит по сайтам и собирает инфу и отсылает на наш сервер.

Если пропал интернет, то мы в JS должны по таймауту отслеживаем нажатия кнопок и уведомляем о том, что проблемы с подключением к серверу.

REST запросы:

/

/find

/rem

/addstore

/user

/user/find

/user/mem

/user/addstore

Функции на JS:

Проверить есть ли куки и отправить запрос на сервер: или считать данные по ID, или создать нового юзера.

Найти цены на товары

Функции на Java

Проверить сессию и регистрацию

Основные фунции движка:

Автодополнение в строке поиска

Занесение в БД

Отбор из БД:

Отбор из

Минимальный набор реализовать, остальное потом.

Работа блока autocomplete связывание с запросом к БД и получение данных в формате JSON.

Элемент HTML > POST запрос с текстом > аннотация в doPost > вызов класса в model.

Может быть GET запрос <http://example.com/findforms/?term=foo>

или POST запрос

<http://example.com>