Overview

Freshipes is an online recipe application aimed at bridging the gap between looking at recipes and buying ingredients.
Design Goals

• Replace ‘print ingredients’ with buy ingredients.
• Create a better search for recipes.
• Recommend recipes to a user using reviews and ratings.
Crawler/Scraper

Images:  http://recipes.wikia.com/wiki/
Reviews:  http://allrecipes.com

Using Scrapy (open source screen scraping and web crawling framework) and other python utilities

Connect data dump from wikia with images
  •  over 2000 images
allrecipes had huge volume of reviews and was very well structured
  •  over 150000 reviews (rating, date, user name, recipe)
Massaging the data

- Used a mediawiki db dump from recipes.wikia.com
  - Took them a while to create a new dump (Dec. 03)
- A good amount of python glue to translate between mw xml schema and our ad-hoc one.
  - Designed to be easy to dump to a tsv for import.
  - Proved to be extensible enough for our needs.
  - Took awhile to find a good library for parsing mw markup fragments down to html.
- In the end this approach didn’t prove to be much easier than just scraping all the data, though the product is probably a little better.
Infrastructure

• What we said:
  • Django on apache/mod_wsgi.
  • Design site to scale.
  • Learn something new.

• What we did:
  • LAMP (PHP, Python)
  • Threw any notion of scaling under a bus.
  • Stuck to stuff we had any level of familiarity with.

• What we learned:
  • Ajax + JQuery
  • Python for web development (Django)
Surprises!

• What we thought:
  • Herp derp....we have a lot of time.
  • wiki means clean data.
  • Mediawiki had a sane category/tagging system.

• Reality:
  • You Don’t.
  • It takes wiki experts to maintain a wiki. People adding recipes are *not* experts.
    • Bad image links, bad quality images, malformed mediawiki markup, inconsistent markup, unmarked stubs, etc.
  • Mediawiki has an infuriatingly useless and completely counter-intuitive way of ‘categorizing’ pages.
    • Fortunately it yielded readily, for our purposes, to a lexicon based IE system.
Lessons learned

• Time is not on your side—start projects early!
• Building a site is a lot easier when you throw out scalability and maintainability
  • fast, cheap, or good pick two one in action.
• Start projects early!
Features

- Search recipes on tags and titles
- Tag categorization (lexicon based IE)
- Recommendation with Slope One
  - Easy enough, (< 50 lines of python for an offline processing version).
  - First feature to get cut.
- Purchase ingredients in Amazon Fresh
- Top rated recipes and New recipes
- Add new comments to a recipe via Facebook
  - Wimped out on doing a real login system.
  - Easy feature to cut.
Questions?