Cultural Properties Browser: Design and Implementation

Matthew Peterson

2015-11-18

Task

- "Design and implement a web application that uses a backend language and runs on the **LAMP stack** that allows end users to browse a collection of approximately 20,000 images representing approximately 5,000 cultural properties from nearly 500 sites within Oregon."

- "Provide a link to
  - a sandbox where your application is running
  - as well as a link to your profile that includes the GitHub repository with the code used for the implementation."

LAMP and Other Technologies

- **L**inux Operating System
  - (Ubuntu LTS v14.04.3)
- **A**pache Web Server v2.4
- **M**ySQL Database Server v5.5
- **P**HP programming language v5.5
  - Using PHP Data Objects (PDO) to access MySQL
- LightVC
  - "lightweight model-view-controller (MVC) framework without the model" written in PHP
- Bootstrap
  - Front end "responsive" framework of HTML/CSS/JS templates
- Dynamic Dummy Image Generator
  - PHP program to create images containing text

GitHub Implementation

https://github.com/petersm3/browser

GitHub Issue Tracking and Milestones

https://github.com/petersm3/browser/issues

Reference – buildingoregon.org

- "Building Oregon is a collection of approximately 20,000 images representing approximately 5,000 cultural properties from nearly 500 sites."
  - http://library.oregonstate.edu/building-oregon

Browser Features

- Faceted (search/navigation/browsing)
  - "technique for accessing information organized according to a faceted classification system, allowing users to explore a collection of information by applying multiple filters"
    - https://en.wikipedia.org/wiki/Faceted_search
- Speed in querying and returning results at all stages of search.
  - Store a large set of <u>arbitrary</u> metadata as I am not a subject matter expert in cultural properties.
- Web Content Accessibility Guidelines (WCAG) 2.0 compliance.
- Responsive design for phone, tablet, and desktop.

Browser Interface
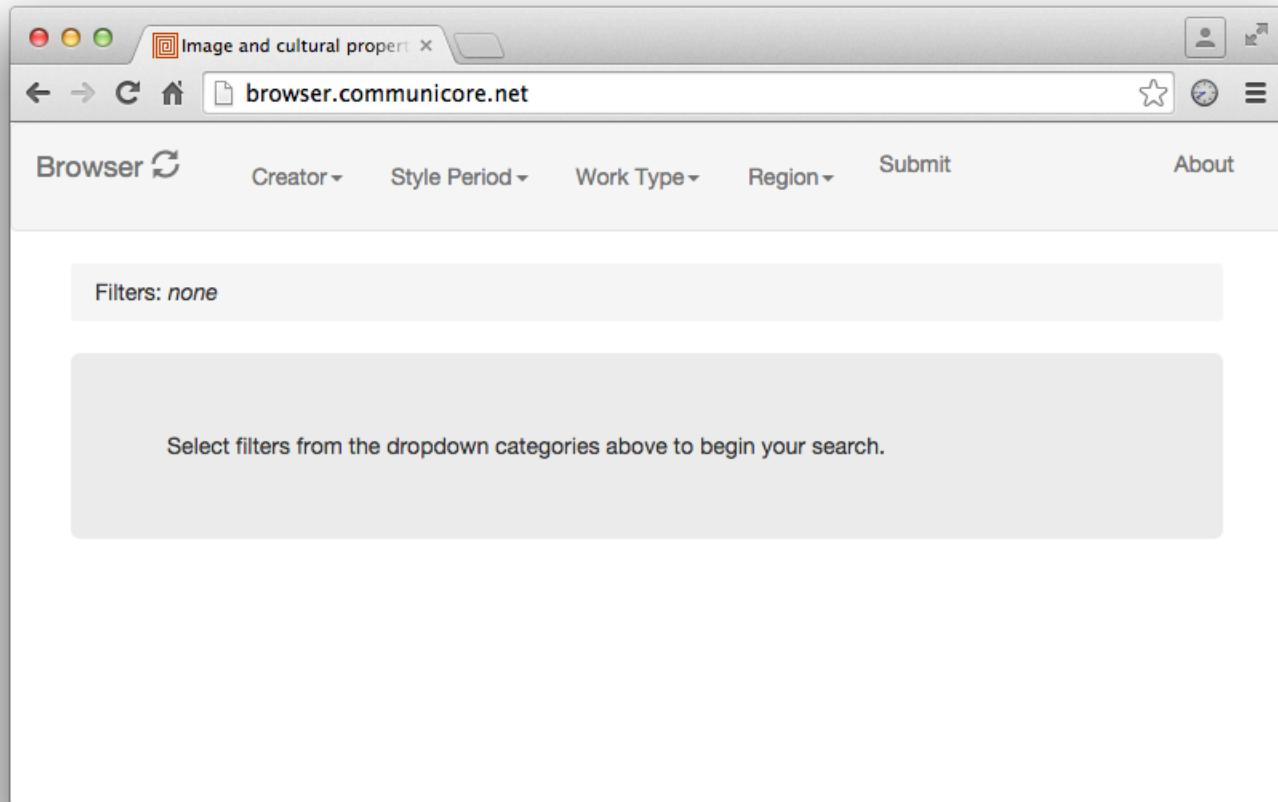
- Four pre-defined categories of filters to select from:
  - "Creator," "Style Period," "Work Type," and "Region"

Browser Faceted Navigation
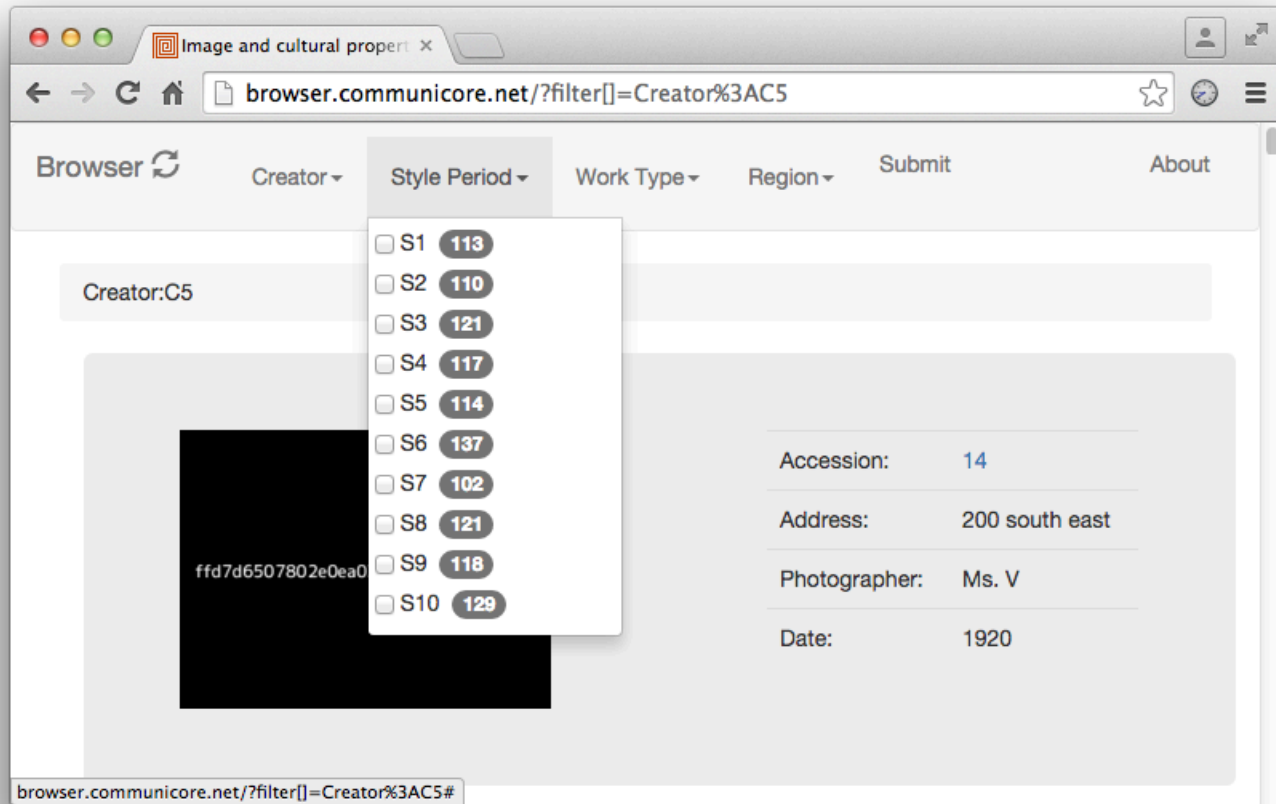
- Each category contains 10 sub-categories of (arbitrary) filters.
  - Database populated with 20,000 accessions (images).
  - Accessions randomly assigned to 4 sub-categories for each category.

Browser Image with Metadata
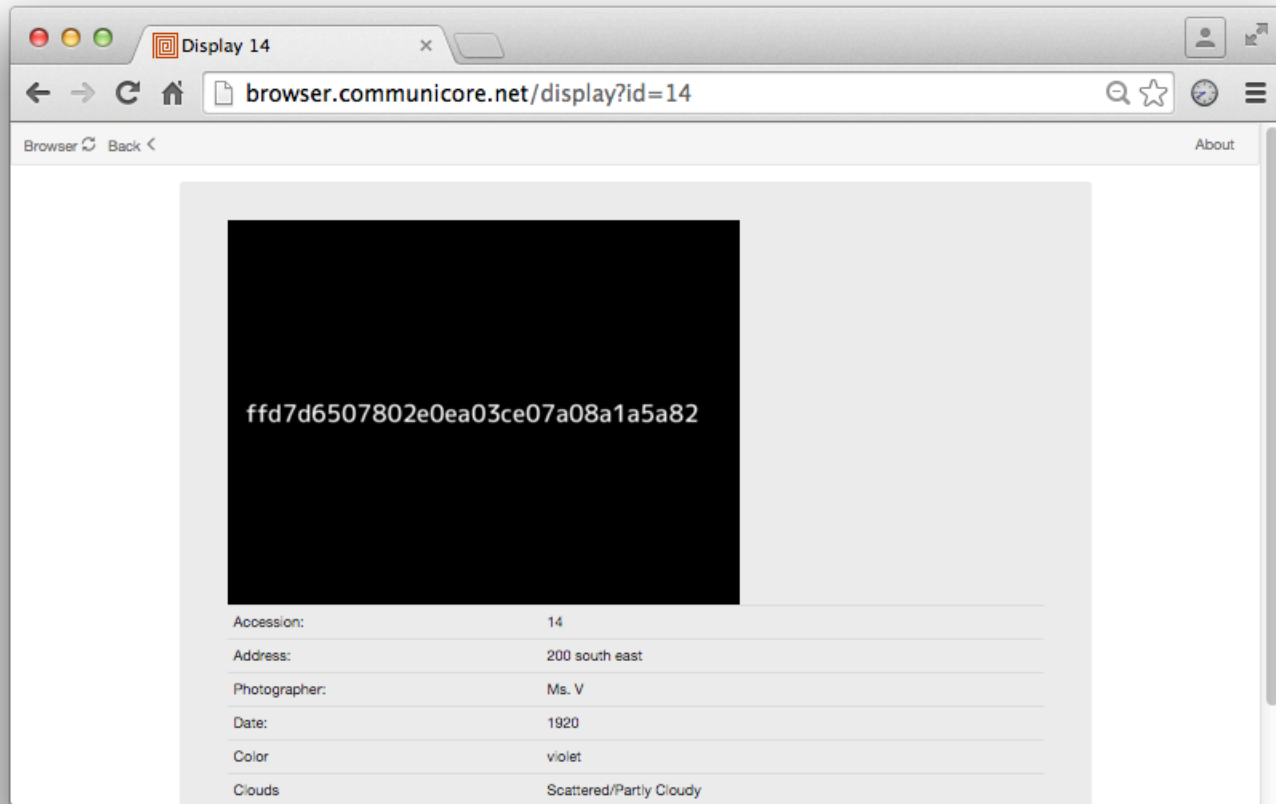
- Images generated with Dynamic Dummy Image Generator.
  - Easy way to generate 20,000+ unique images in real-time.
  - Served via a simulated Content Delivery Network (CDN).

MySQL Database Schema

categories
- 🔑 id INT(2)
- ◇ priority INT(2)
- ◇ category VARCHAR(256)
- ◇ subcategory VARCHAR(256)
- ◇ comment VARCHAR(1024)
- Indexes ▶

properties
- 🔑 id INT(7)
- ◇ image VARCHAR(32)
- ◇ street_address VARCHAR(255)
- ◇ photographer VARCHAR(255)
- ◇ date VARCHAR(255)
- Indexes ▶

filters
- 🔑 id INT(7)
- ◆ fk_properties_id INT(7)
- ◆ fk_categories_id INT(2)
- Indexes ▶

attributes
- 🔑 id INT(9)
- ◆ fk_properties_id INT(7)
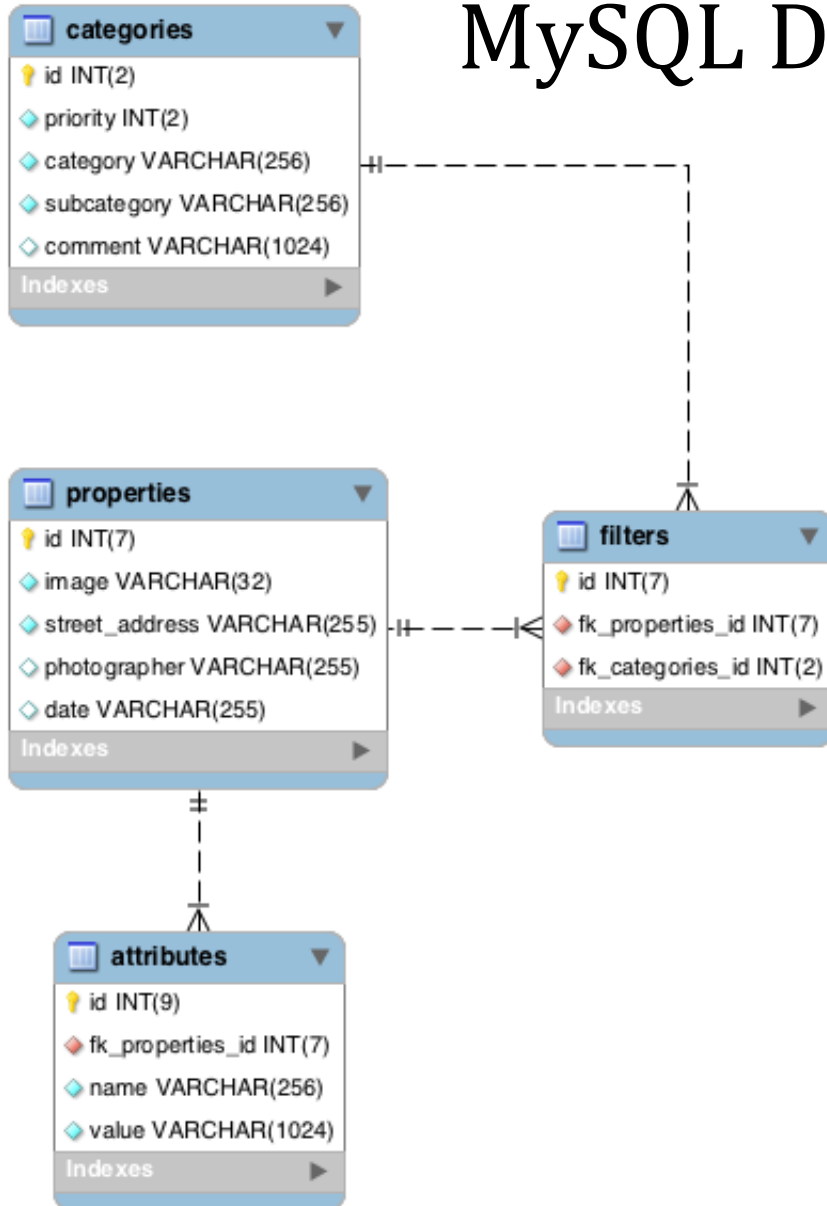- ◇ name VARCHAR(256)
- ◇ value VARCHAR(1024)
- Indexes ▶

- `categories` table to helps build navigation dropdown of categories and sub-categories
- `filters` table maps categories and sub-categories to each accession (image)
- `properties` table defines each of the 20,000 randomly generated accessions (images)
- `attributes` table stores additional, arbitrary metadata for each accession

Challenges Encountered

- I am not a subject matter expert on cultural properties so I opted to build a browser for an arbitrary set of metadata.

- Faceted navigation interface took the most time to develop.

- HTML validation incomplete (issues with a <div> and [])
  - https://github.com/petersm3/browser/issues/15

- Time constraints; there's always more features to add:
  - Free form text search
  - Controlled vocabularies for metadata
  - Alphabetical and other types of sorting
  - User selectable pagination ranges